NASA Contractor Report 187197

Probabilistic Structural Analysis Methods (PSAM) for Select Space Propulsion System Components (3rd Annual Report)

Southwest Research Institute
San Antonio, Texas

October 1991

Prepared for
Lewis Research Center
Under Contract NAS3–24389

NASA
National Aeronautics and
Space Administration

TABLE OF CONTENTS

TABLE OF CONTENTS
(Continued)

TABLE OF CONTENTS
(Continued)

TABLE OF CONTENTS
(Continued)

LIST OF TABLES

LIST OF FIGURES

LIST OF FIGURES
(Continued)

LIST OF FIGURES
(Continued)

LIST OF FIGURES
(Continued)

1.0 PSAM PROJECT OVERVIEW

1.1 Introduction

This Annual Report summarizes the work completed during the third year of technical effort on the referenced contract. Principal developments continue to focus on the Probabilistic Finite Element Method (PFEM) which has been under development for three years. Essentially all of the linear capabilities within the PFEM code are in place (Section 2.0); most have been validated (Section 6.0). Major progress in the application or verification phase has been achieved for the PFEM and is reported in Section 7.0.

Additionally, the EXPERT module architecture has been designed and partially implemented, as reported in Section 4.0. EXPERT is a user interface module which incorporates an expert system shell for the implementation of a rule-based interface utilizing the experience and expertise of the user community. EXPERT has been substantially modified from the Second Annual Report to incorporate a C-language expert system shell, CLIPS, written at NASA Johnson Space Center. The use of the C-language allows for an effective interface to a variety of needed Fortran utility subroutines. These subroutines perform a variety of operations on data sets used in the input and control of the PFEM and other modules that form the bulk of the user interface.

The Fast Probability Integration (FPI) algorithm continues to demonstrate outstanding performance characteristics for the integration of probability density functions for multiple variables (Section 3.0). Several minor enhancements to the algorithm are reported. Additionally, an enhanced Monte Carlo simulation algorithm has been developed at the University of Arizona under Professor Wirsching's direction. A variety of numerical strategies were investigated in the process and are detailed in Appendix C.

1.2 Probabilistic Finite Element Method (PFEM)

The finite element algorithms are broadly classed in terms of the standard displacement method and as a mixed method with iteration for nodal equilibrium. Within each method the user has access to a variety of element types, as developed in the first two years of the contract effort.

During the past year the PSAM project has implemented two new element types within the PFEM module. The MARC team, under the leadership of Drs. Joop Nagtegaal, S. Nakazawa, and Mr. Joao Dias, has implemented an advanced shell/plate element with the ability to handle through-thickness gradients. The element is an eight-noded solid element with assumed strain freedoms. Shell/plate behavior has been achieved in terms of a large aspect ratio capability for the element by the proper selection and tuning of the assumed strain terms.

The second new element is the sixteen-node hybrid (assumed stress) element developed under the direction of Dr. Satya Atluri and his staff at the Georgia Institute of Technology. Again, the element has surface nodes and is capable of aspect ratios approaching shell/plate requirements.

1.3 Probabilistic Boundary Elements (PBEM)

The focus during the past year has been on the development of a proper formulation strategy to permit the extension of an existing boundary element code to the probabilistic context. The selected BEM code for that development is the BEST3D code developed under NASA HOST funding in an effort directed by Drs. Banerjee and Wilson; much of that coding was accomplished by Dr. Raveendra, now working on the PBEM implementation.

The PFEM strategy is to compute structural solutions for perturbed states of the random variables using an iteration algorithm. In this algorithm, the perturbed variables are shifted to the right-hand side of the system equations, and the perturbed solution obtained by iterating with the reduced stiffness matrix serving as a pre-conditioning matrix.

The PBEM investigation has reviewed the strategies available for the generation of perturbed solutions. Since the BEM formulation is in terms of surface variables, it was at first most natural to think of a direct means of computing geometry perturbations in analytical rather than numerical terms. While technically feasible, the analytical approach appears to involve substantially more cost of implementation than the numerical approach; thus the latter approach was selected.

The use of BEM formulations for high temperature gradient problems in turbomachinery requires a treatment of volume terms associated with non-steady thermal strains, inhomogeneous material properties, and plastic strains. The usual treatment of these terms is through volume integrals requiring discretization of the body volume. Recent research in the BEM community has identified the potential use of surface-based interpolation functions for these volume integrals. The PBEM formulation has been based on the use of such surface interpolators. Perturbations are then performed in terms of surface data, even for internal variables, by this strategy.

1.4 Code Validation and Verification Studies

Code validation and verification are critical elements in the PSAM effort. Code validation is a task to establish the ability of the integrated analysis and probabilistic modules to generate the "exact" solution to simple problems, amenable to independent analysis. Code verification is to demonstrate the ability of the PSAM codes to generate meaningful probabilistic

analysis results for each of four SSME component analyses. The verification analyses, therefore, generally involve large modeling problems and loading conditions that preclude comparison to analytical results.

The validation studies have made significant progress in the past year in terms of the number and diversity of the problems that have been solved. A standard format for the validation problems has been established that will facilitate the evaluation and replication of these results by other users.

The validation results have identified code errors and shortcomings that have been resolved. More importantly, these problems have given significant insight into the operation of the PSAM codes for various types of modeling problems. These insights are being used to develop rules in EXPERT that will ease the user burden for these classes of problems.

Additionally, the validation problems have provided critical technical insight into the nature of probabilistic analysis results. In particular, the results have all shown that, while the deterministic modeling answer my be off from the known solution, the distribution of the probabilistic solution is highly accurate. Thus, by calibrating the model at the deterministic solution point, the PSAM algorithm is able to correctly predict the distribution of the results relative to the deterministic solution. This derives from the observation that the PSAM algorithm is based on the use of sensitivity data from the perturbation algorithm; sensitivity data is seen to be quite accurate so long as the physics of the problems has been properly modeled.

The first major verification problem is nearly complete. The PSAM algorithms have been applied to a turbopump blade analysis. The random variables include geometry and material properties for the static analysis. Current work is applying random loading conditions and analyzing the dynamic response characteristics of the blade.

Dr. Rajagopal of Rocketdyne has made major contributions to the PSAM effort in the verification task. He has identified numerous code problems which have been fixed as well as developing effective graphics interfaces for the PSAM results which facilitate the interpretation of the data.

1.5 Planned FY88 Technical Effort

Two major new tasks are underway in the current Fiscal Year (the fourth year of the project). The first is the implementation of Probabilistic Approximate Structural Analysis Methods (PASAM) for selected components. The PASAM algorithms have been defined for each of the four components. The PASAM algorithms will be based on the observation made from the validation examples that the distribution of the solution can be accurate to within a deterministic calibration value, if the physics of the random variables are

properly accounted for. Thus, each of the four problem formulations will focus on the definition of critical response variables, and on the definition of the role of each of the random variables.

Simplified mechanics models will be generated to estimate the required solution variable dependence on the random variables. It is expected that the deterministic solution will be crude and in error. It is assumed that a calibration analysis or an experimental result exists for defining an accurate deterministic solution. PASAM will generate distributional results, normalized to the deterministic solution. Thus, the analyst will be able to rapidly determine the sensitivity of the response variable to the random variables, as well as to predict the overall uncertainty in the design response variable. It is likely that this version of the PSAM capability could be PC-based.

The second new task is the development of a Level III probabilistic material behavior model. The goal is to predict random stress-strain curves that derive from considerations of basic material mechanism behavior or appropriate phenomenological models from zero load to ultimate load. Consideration will be given to basic probabilistic variables for describing materials (grain sizes, defect structures, orientations, temperatures, etc.) such that the simulations can show the dependency of the response stress-strain curve character to the independent random variables. Interactions between mechanisms and dependencies between certain random variables is to be included.

2.0 NESSUS FINITE ELEMENT CODE DEVELOPMENT

2.1 Introduction

The NESSUS finite element code is being developed by MARC Analysis Research Corporation as part of the probabilistic structural analysis (PSAM) effort, coordinated by Southwest Research Institute for the NASA-Lewis Research Center. The objective of this effort is to provide an advanced analysis capability by combining the versatility of a modern finite element code with the latest developments in the area of probabilistic modeling and structural reliability. Special attention was devoted to the efficiency and generality of the algorithms adopted in order to make the code usable for the analysis of realistic engineering problems which are representative of typical SSME applications.

2.1.1 Status at End of FY '86

During FY '86 the NESSUS finite element code gradually evolved from a purely deterministic finite element code into a basic probabilistic analysis code. Version 1.1 of the NESSUS code was released to all members of the PSAM team in March '86 and was being extensively exercised at all sites by the end of FY '86. NESSUS 1.1 allowed linear elastic and eigenvalue analysis of structures with uncertain geometry, material properties and boundary conditions, subjected to a random mechanical and thermal loading environment. Probabilistic analysis with this version of the code was limited to a single increment of elastostatic or dynamic eigenvalue analysis, using the displacement formulation, and with no initial strain and/or stress effects.

Initial experience with NESSUS 1.1 by the PSAM team members indicated the need for several enhancements to be provided with the second year code. The desired enhancements included:

- A faster equation solver using profile storage.
- The ability to update an existing perturbation database with results obtained in multiple runs.
- A "smarter" elastostatic perturbation algorithm, able to bypass most redundant or unnecessary computations.
- The ability to reformulate the unperturbed solution at a point other than the mean.

- A more flexible set of integration schemes for strain recovery and projection, accommodating collapsed element configurations.
- More user-friendly input of material properties for certain classes of anisotropic materials, allowing these parameters to be random.
- The need for an enhanced 3D continuum element which could be degenerated to a high aspect ratio to model plate and shell-like structures.
- An algorithm for defining surface pressures on a nodal basis.
- Performance improvements on the subspace iteration algorithm used for modal analysis.
- The explicit addition of the second tensor invariant for strains and stresses in the perturbation database.

All of the above were being addressed in the development version of NESSUS by the end of FY '86, and were to be included in NESSUS 1.5 and 2.0, released to the members of the PSAM team on December '86 and February '87, respectively.

An important feature lacking in these earlier versions of NESSUS was the ability to introduce random initial strain and/or stress effects in the analysis. These can be rather significant in probabilistic analysis of rotating machinery when stress stiffening effects, due to larger centrifugal loads dominate the response. The solution strategy involved carrying initial stress terms for each perturbed problem across two increments in a consistent manner, and had been demonstrated in a special version of NESSUS in October '86. However, the general multi-increment perturbed problem capability was not available as a standard feature of NESSUS until the release of version 2.5 in September '87.

The planned extension of the perturbation algorithms in NESSUS to multi-increment, inelastic problems raised some important issues involving the internal data representation and the choice of a solution strategy. In particular, a decision had to be made regarding whether to pursue: (a) a pure displacement-based formulation allowing the internal storage of the element stresses and strains on an integration point basis, or (b) an MHOST-type mixed-iterative formulation allowing the storage of all stresses and strains on a purely nodal basis. By the end of FY '86, a decision had been made to pursue the latter.

Although it necessarily involves the adoption of a less mature finite element technology, the decision to pursue the mixed-iterative approach allowed the use of a nodally-based strain recovery scheme as defined by the NASA Statement of Work. This approach also lends itself to a more elegant implementation of the inelastic perturbation analysis algorithms and a cleaner interface to the external perturbation database. By eliminating the need to remember the stress/strain history at the element integration points, the amount of data stored in the perturbation database is reduced, which helps keep the database files within a manageable size. The adoption of a mixed-iterative strategy allowed large portions of FORTRAN code to be shared between the NESSUS and MHOST codes, facilitating the cross-transfer of new technology between these two codes. Nevertheless, due to the computational economy achievable with the displacement formulation in linear elastostatics, the option of invoking the displacement method for perturbation analysis of linear problems will be retained in the NESSUS finite element code.

2.2 Code Deliveries During FY '87

NESSUS 1.5 was released to SwRI, Rocketdyne and GIT in December '86. The objective of this limited release was to allow these subcontractors to exercise the code in order to identify any outstanding problems that needed to be addressed prior to the scheduled delivery of the second year code in February '87. This version of NESSUS addressed most of the needs identified while exercising NESSUS 1.1 on representative engineering problems. NESSUS 1.5 also provided for the first time the ability to conduct perturbation analysis on problems based on a mixed-iterative formulation, although it lacked the fine control over iteration tolerances that would be desirable for the effective use of this strategy.

The second year code, identified as NESSUS 2.0, was delivered to all members of the PSAM team in February '87. The main feature introduced with this version was an enhanced 3D continuum element based on an assumed strain field formulation and designed for improved accuracy in bending problems. This element can be degenerated to a high aspect ratio in order to reproduce thick plate and shell-type situations, and provides for surface pressure definition and strain recovery on a nodal basis, as defined in the NASA Statement of Work.

NESSUS 2.5 was released to the members of the PSAM team in September '87. New features introduced with this version include the ability to carry perturbation results across multiple load increments, finer control over iteration tolerances for use with mixed-iterative and eigenvalue problems, and a full library of assumed strain continuum elements with enhanced bending behavior. This version can accommodate random initial strain and stress fields, in order to capture the uncertainties in the stress stiffening effects governing the response of rotating machinery subjected to large centrifugal stresses.

2.3 Extension of NESSUS/FEM to Mixed Method and Incremental Analysis

The objectives of the PSAM effort include the development of probabilistic finite element methods for handling not only linear problems but also problems involving nonlinear material and geometric response. A successful strategy for achieving these goals will require: (a) the development of the means for tracking several perturbed solution paths across multiple increments, and (b) the ability to compute accurate response sensitivities for problems which have not been or cannot be iterated to a very high accuracy. Both issues were addressed during the past year as part of a strategy for extending NESSUS/FEM to the mixed method and incremental elastostatic analysis. These extensions involve data manipulations which are very similar to those needed for mildly nonlinear problems, and this development may be regarded as the first step towards the extension of NESSUS/FEM to material and geometry nonlinear situations.

As stated above, the desire to rely on a purely nodal data representation for stress and strain for inelastic problems naturally led to the adoption of a mixed finite element formulation [1] expressed in terms of nodal displacement, stress and strain. A practical approach for the solution of the mixed problem was developed under the auspices of the HOST program at NASA/LeRC and implemented in the MHOST code. The MHOST implementation relies on an iterative strategy to recover the mixed solution, using the displacement method solution as the iteration preconditioner. With this approach, the introduction of stresses and strains as mixed variables does not significantly increase the problem size, since only a matrix with the size of the number of displacement degrees of freedom needs to be factorized.

In the analysis of inelastic problems, the mixed method can be used effectively by combining the nonlinear iteration with the recovery of the mixed solution in the same iterative loop. Since in typical nonlinear problems the residuals are not iterated to within machine accuracy (the residual load correction term automatically carries it forward into the next increment), the mixed-iterative approach does not require a number of iterations that is significantly different from that used with the displacement method.

That is not the case in the analysis of linear elastic problems, since the direct solution of the displacement equations will yield a residual force vector that merely reflects machine round-off in the multiple iterations, even for a linear problem, in order to recover the mixed solution from the displacement result. It should be noted, however, that the greatest improvements in the strain and stress solutions obtained with the mixed method occur in the first few iterations. Hence, for many problems, it is rather uneconomical to attempt to iterate the "mixed residual" to a very low value.

Initial experiments with the use of the elastostatic perturbation algorithms using an MHOST-type mixed-iterative formulation have demonstrated the feasibility of the approach. However, problems will arise whenever the magnitude of the imposed perturbations result in a change in residuals that is smaller than the residual carried over from the unperturbed problem. It should be noted that a similar situation will be encountered when perturbing a nonlinear analysis performed with the displacement formulation, since in nonlinear problems the residuals are not usually iterated to a very small value (just as with the mixed-iterative formulation).

The solution involves separating the residual load component induced by the perturbation from the residual load vector carried over from the unperturbed problem. This strategy has been described as an "equilibrium shift" and amounts to computing the displacement update for the perturbed problem using

$$\hat{d}^{n+1} = \hat{d}^{n} + K^{-1}\left(\hat{f} - \int_{\Omega} \hat{\hat{B}}\hat{Q}^{n} - r\right) \qquad (2.1)$$

where a carat is used to denote the perturbed quantities and r is the uniterated residual load vector from the unperturbed problem. This approach provides an effective method for computing the sensitivity of a solution which has not been iterated to a very small residual.

The basic mechanism used to perform "equilibrium shift" is implemented in NESSUS 2.5. In order to use "equilibrium shift" effectively, it is necessary to manipulate the iteration controls differently for the unperturbed and perturbed solutions. The recommended approach involves the generation of a mixed unperturbed solution in which the iteration tolerances are relaxed in order to achieve convergence with only a few steps of mixed strain recovery. This represents a relatively inexpensive way of improving the smoothness of the stress and strain solutions. All subsequent perturbation problems are then iterated to a finer tolerance, which is imposed only on the component of the residual load induced by small deviations of the random variables from their unperturbed values. The net effect is to prevent the response sensitivity calculation from being lost in the noise present in the unperturbed solution. When used in this way, the efficiency of the perturbation algorithm using the mixed-iterative formulation can be made to approach that of the displacement method, while retaining all the desirable features associated with the mixed formulation.

The extension of NESSUS/FEM to incremental analysis involved the development of a mechanism for tracking several perturbed solution paths across multiple increments. In general, this will include the ability to recover the total displacements, stresses, elastic and plastic strains, and any other state variables from the converged solution for the current perturbation at the previous increment. By relying on a mixed-iterative formulation with all state variables defined on a nodal basis, it is possible to use the current structure of the perturbation database to store and recover these quantities. The same holds for linear elastic analysis using the displacement formulation, even if multiple load increments are present. The data structure currently implemented in the perturbation database would not be adequate for the use of the displacement formulation for inelastic problems, since that would require the availability of any history-dependent state variables (e.g., plastic strains) on an element integration point basis. This is one reason why the displacement method is not being pursued for inelastic problems within the PSAM effort.

The simplest type of analysis involving a consistent tracking of perturbed solutions across multiple increments is related to the introduction of random initial stress and strain fields. Initial stress effects account for the change in lateral deflection and natural frequencies when a turbine blade is subjected to large centrifugal loads. If there are uncertainties in the rotation speed, geometry, material properties, etc., these will introduce uncertainties in the initial stress field, which can be estimated by a perturbed elastostatic analysis. In the following increment, each perturbed initial stress field obtained in the previous increment is used to compute the stress stiffening effects for the corresponding perturbation. It is important that the bookkeeping is done correctly, so that the change in initial stress resulting from perturbing a given random variable is accounted for when the same variable is perturbed again in the following increment. In a similar manner, it is possible to include the effect of a random initial stress field computed in a probabilistic elastostatic problem on a subsequent random eigenvalue analysis. Random initial strain fields (induced, for instance, by a random temperature field) are handled in exactly the same way.

As described above, in order to minimize in-core data storage requirements, NESSUS/FEM utilizes the perturbation database for temporary storage of the perturbed initial strain and stress fields. Hence, for analyses involving stress stiffening effects, the perturbation database size may have to be expanded in order to accommodate the generalized initial stress field. However, in order to keep the database size as small as possible, these additional quantities will not be stored unless it is clear that they are needed for the type of analysis in progress. It should be noted that earlier versions of the ENCODE and DECODE utilities (from NESSUS 2.0) remain compatible with the current format of the perturbation database. The same holds true for the perturbation database interface to NESSUS/FPI.

2.4 Advances in Element Formulation

Within the past year, several members of the PSAM team have expressed a desire to develope advanced element technology tailored to address specific SSME applications in a more effective manner. Many of the components addressed with this effort can be characterized either as a slender continuum or as a very thick, variable thickness shell using current finite element technology. An accurate solution can usually be obtained by using continuum elements. However, the modeling of slender shell-like components as a 3D

continuum requires a very fine mesh and involves considerable computational and modeling effort. This approach is often too expensive for standard design practice (see Figure 2.1). On the other hand, if shell elements are used, the computational effort is reduced by some degree of accuracy and resolution is often sacrificed. This is often the case if the structure exhibits strong curvature (of radius less than five times the thickness of the shell), large thickness variations or very localized thermo-mechanical loading. Also, the stresses in the neighborhood of shell intersections and connections are not accurately calculated.

Many heuristic rules have been developed for the use of shell elements in similar problems. This is frequently done by selecting an "effective" thickness near the discontinuity or by coupling the intersections in special ways. Nevertheless, it would be useful to have 3D elements with which such problems could be modeled effectively and accurately. In principle, continuum theory should always be able to represent the "exact" solution.

However, regular continuum elements often lack the appropriate deformation modes to model shell-like structures in a satisfactory way. This was first observed by Ahmad and Zienkiewicz [3] in the development of the classical 8-node thick shell element. The problem was partially overcome by using a reduced integration formulation. Similar ideas were used later on in the development of thick shell elements. These elements frequently resort to the use of special interpolations for the transverse shear terms in order to retain the ability to accurately model the bending behavior of shells. These include the Heterosis element of Hughes and Tezduyar [4] and the 8- and 9-node thickness element proposed by Hinton and Huang [5]. It can be argued that similar interpolations could be used for the strain and/or stress field within continuum elements. Hence, one should be able to design continuum-like elements that perform well when degenerated in one direction to model shell-like structures.

There are several known strategies for constructing continuum elements with enhanced bending behavior (see Figure 2.2). One of the first attempts employed the use of selective integration [6]. The original element was very successful in the rectangular configuration when aligned with the global coordinate system, but behaved poorly otherwise. The formulation of Kavanagh and Key [7] cured the problem by introducing a local cartesian element coordinate system and thus making the shear term invariant with respect to a

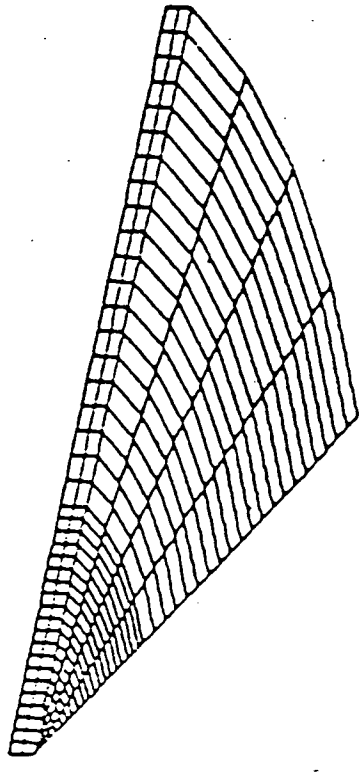
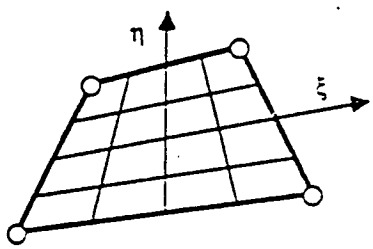
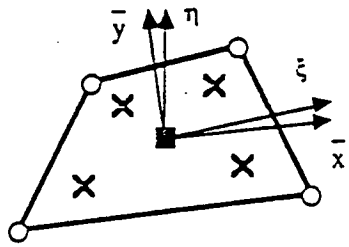


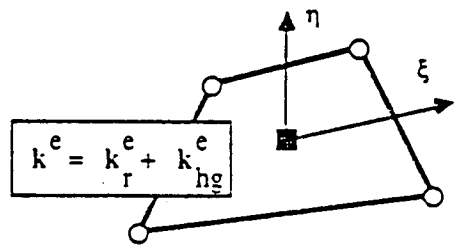
Figure 2.1 Localized Thermal and Mechanical Effects Crucial to the Analysis of Components Such as This Blister Specimen Model Cannot be Adequately Captured Using Simplified Shell Models



The Taig Quadrilateral
Taig and Kerr (1964)

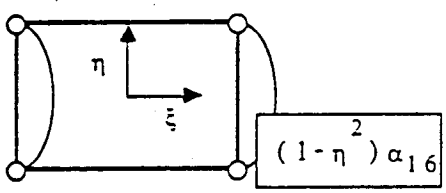


Selective Reduced Integration
Kavanagh and Key (1972)

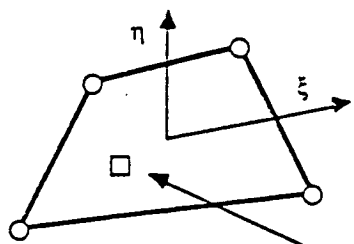


Full Reduced Integration
With Hourglass Control
Kolsoff and Frazier (1978)
Flannagan and Belytschko (1981)

SOME FORMULATIONS YIELD EQUIVALENT ELEMENT STIFFNESSES



Incompatible Modes
Wilson (1973)
Taylor and Wilson (1976)



Assumed Stress Hybrids
Pian and Sumihara (1984)

Five Independent
Strain Parameters

Figure 2.2 The Evolution of Low-order Continuum-type Elements Towards Improved Accuracy in Bending Problems

change of the global coordinate system. This formulation is not easily extended to anisotropic problems, in which the shear terms may be coupled with the direct stress components. A related approach involves the use of full reduced integration with the addition of hourglass control modes which are designed to enhance bending behavior. Examples include the formulation of Kosloff and Frazier [8] and the elements advocated by Flannagan and Belytschko [9]. These elements lend themselves to very efficient implementation and have become quite popular for certain applications. The original library of continuum elements implemented in NESSUS/FEM (element Types 3, 7, 10 and 11) are based on a similar formulation. Another early attempt by Wilson [10] resorted to the addition of two incompatible quadratic "bubble" modes in an effort to reproduce the quadratic displacement field corresponding to "pure bending." However, when the element assumed the form of an arbitrary quadrilateral, it was found to behave erratically and failed the patch test. A cure for the problem was proposed by Taylor and Wilson [11] which is based on the evaluation of the "bubble" function derivatives at the centroid of the element. The resulting element was found to pass the patch test for arbitrary configuration, and is currently implemented in a number of commercial codes. However, these elements are used primarily for linear elastic analysis, since it is not readily apparent how the strains associated with the "bubble" modes should be handled in elastoplastic situations. Recently, Pian and Sumihara [12] have proposed a new element which exhibits excellent bending behavior even for somewhat distorted configurations. The element is an assumed stress hybrid, based on the use of five independent stress parameters to define the state of stress at the interior of the element. The assumed stress approach offers some problems regarding the implementation of plasticity algorithms. This is due to the fact that the most successful plasticity algorithms to date have been strain-driven, and not stress-driven. In particular, the implementation of a stress-driven plasticity algorithm (in itself a major coding task) cannot easily accommodate the perfectly plastic case in the absence of work-hardening effects.

The approach pursued at MARC was aimed at the development of a family of continuum-type elements with enhanced bending behavior and retaining good performance when degenerated to a high aspect ratio. Of course, there are limits as to how far one can carry such a degeneration. For elements of length 1 and thickness t, the bending stiffness is of order $0(t^3/1^2)$, both the

membrane and transverse shear stiffness are of order $O(t)$, and the direct transverse stiffness (change of thickness) is of order $O(l^2/t)$. Hence, for numerical reasons, it does not appear desirable to degenerate these elements to an aspect ratio $t/l < 0.01$, which would cause the loss of more than eight digits accuracy. This should not present a major problem, since the primary application of these elements would be for thick shell-type situations, in which an element aspect ratio $t/l < 0.1$ should be adequate. The elements were constructed using an assumed strain formulation. The basic strategy involves the identification of a set of independent stress modes representing the desired element behavior, followed by the construction of a corresponding set of strain models which, under appropriate conditions (for any isotropic material or particular orthotropic material orientation), will yield the desirable stress modes. This allows the formulation of an element which is based on a strain-driven constitutive algorithm, and can be readily implemented within the existing code framework. The strain modes are used to interpolate the strains within the element and are related to the displacement gradients by a weak variational form. All assumed strain modes are expressed in terms of a local element cartesian coordinate system obtained by polar decomposition of the isoparametric mapping at the centroid of the element. This strategy not only simplifies the derivation of the assumed strain modes, but also is expected to enhance the robustness of the element in distorted configurations. The stretch tensor obtained in the polar decomposition is used for computing scale factors to make the element computation dimensionless. This was observed to be particularly useful for reducing round-off for very high element slenderness ratios. Although the cost of forming the B-matrix for the assumed strain elements can be as high as 2-3 times that of a standard isoparametric element, the increase in cost per element is frequently offset by the ability to use a much coarser mesh in bending-dominated problems.

The library of assumed strain elements implemented in NESSUS/FEM includes 4-node quadrilaterals for plane strain, plane stress and axisymmetric problems, and an 8-node solid element for modeling three-dimensional continua. The current implementation can be used with either the displacement or the mixed formulation and supports different integration rules for strain projection and residual recovery. However, due to the unconventional nature of the element formulation, the integration rule for the element stiffness

computation is fixed. These elements can be collapsed into triangles, wedges and tetrahedra, in accordance with the rules implemented for other continuum elements.

In addition, an algorithm has been developed to allow the nodal definition of pressure loading on 2D and 3D continuum meshes (see Figure 2.3). The aglorithm is based on a nodal assembly of tributary areas at each node in such a way that a unique outward boundary normal vector is defined at each surface node. These normals define the effective surface orientiation and the direction of the applied pressure at the node. The basic concept is depicted in Figure 2.3 and involves the following steps:

1. Apply unit pressures to all faces of each element.
2. Compute the corresponding nodal loads.
3. Assemble the element force vectors.
4. The assemble vectors cancel-out at all internal nodes.
5. The actual nodal forces are obtained by multiplying the outward boundary vector by the negative of the nodal pressure value.

For small deformation problems, this operation is carried out only once, during the first element assembly loop, and the resulting boundary normals are used to compute consistent pressure loads throughout the analysis.

In probabilistic finite element problems with uncertain geometry and nodal pressure definition, the boundary normals are recomputed for the perturbed configuration at every geometry perturbation. Hence, if a geometry perturbation results in an increase of the surface area exposed to pressure loading, the corresponding increase in equivalent nodal forces is automatically accounted for in the algorithm. Likewise, in finite deformation problems using an updated Lagrangian formulation, the recomputation of the boundary normals can easily account for the follower pressure effects on the applied loading vector.

2.5 Development of Deterministic Finite Deformation Algorithms

The ability of conduct elastoplastic deterministic finite deformation analysis using a mixed-iterative formulation was introduced in NESSUS/FEM within the time period covered in this progress report. The basic formulation employs a Lagrangian mesh description, with the equations of motion evaluated at the current (deformed) configuration. Often described as an "updated

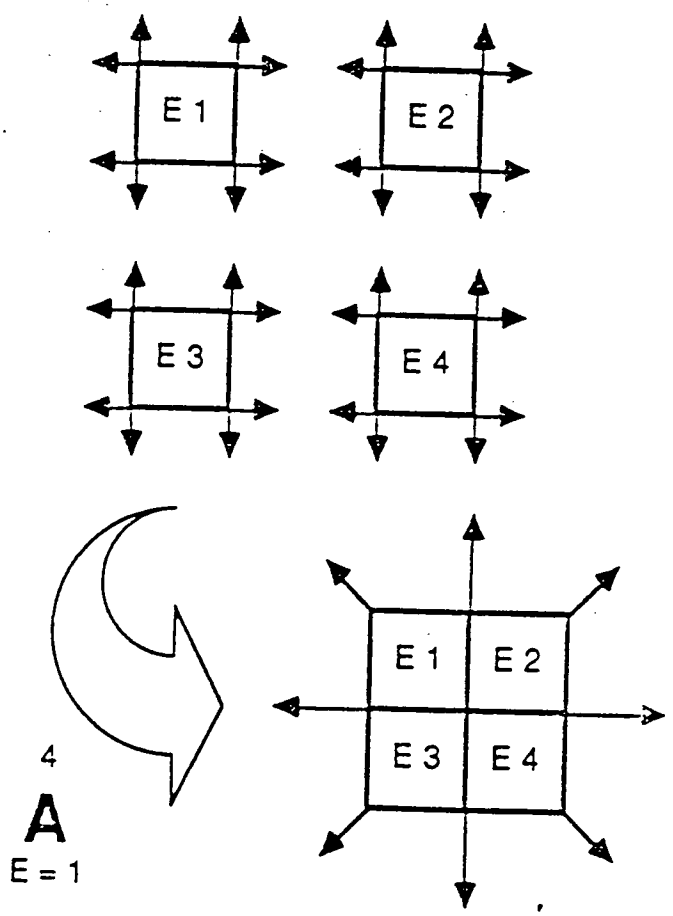


Figure 2.3 An Assembled Boundary Normal Algorithm Allows Nodal Pressure Definition for Continuum-based Models

Lagrangian" formulation [13], this approach offers considerable computational simplicity since, by continually updating the mesh geometry to the current configuration, all matrix expressions can be made to assume the same form as in small deformation theory. The only additional matrices involve deformation gradients and rotation tensors, along with the matrices associated with follower forces.

The follower force components are evaluated using body force and surface traction values at the end of the current increment. The following force matrices associated with change of volume or area are symmetric and easily included in the stiffness reformulation. However, unsymmetric matrices are associated with the rotation of follower forces. In keeping with the philosophy of the mixed-iterative approach, any contributions from unsymmetric matrices are accounted for in the residual load correction term, and recovered by the iterative process. This avoids all the problems associated with the introduction of an unsymmetric stiffness matrix.

Using the mixed formulation, concentrated nodal follower forces enjoy the advantage of having the necessary rotation tensors readily available on a nodal basis as an integral part of the formulation. These nodal rotation tensors are easily obtained by polar decomposition of the nodal deformation gradients.

The constitutive equations for elastoplastic finite deformation computations are based on the use of the Green-Naghdi rate of Cauchy stress and rate of deformation [14]. This rate was chosen for its computational efficiency, and its ability to avoid non-physical oscillatory stress response when used in conjunction with kinematic hardening [15]. The specific rate form for Cauchy stress used in this implementation can be expressed as

$$\sigma^{\nabla} = \dot{\sigma} + \sigma\dot{R}\,R^{T} - \dot{R}\,R^{T}\sigma \qquad (2.2)$$

where R is the rotation tensor obtained by polar decomposition of the nodal deformation gradients. The resulting constitutive equation

$$\dot{\sigma} + \sigma\,\dot{R}\,R^{T} - \dot{R}\,R^{T}\sigma = D\,d \qquad (2.3)$$

where d is the rate of deformation tensor. can be transformed using the rotation tensor to the equivalent form

$$\dot{\sigma}_R = D_R d_R \tag{2.4}$$

where

$$\dot{\sigma}_R = R^T \dot{\sigma} R$$

$$D_R = R^T D R$$

$$d_R = R^T d R$$

For the continuum elements, the constitutive laws are expressed in the global coordinate system, and the above transformations can be utilized directly. Thus, the evaluation of the constitutive equation involves transforming its components from the global to the rotated coordinate system, with the actual evaluation being form-identical to the small deformation case.

By contrast, the constitutive equation for the shell element is expressed in terms of a local Cartesian coordinate system, defined by averaging the normal vectors for all shell elements connected at the node. In finite deformation computations, the local system is continuously recomputed during the geometry update process. This results in a nodal coordinate system which remains normal to the shell surface as the model deforms. Therefore, the local coordinate system in which the constitutive equations are expressed will rotate with the structure.

The finite deformation algorithm implemented for the shell element takes advantage of this fact to avoid additional calculations involving transformations to the constitutive equation. This effectively replaces the rotation tensor in the equations above with a continuously updated global-to-local transformation tensor.

A similar transformation was implemented for the beam element.

2.6 Enhancements to Database Manipulation

The perturbation database format implemented in NESSUS provides considerable flexibility for management perturbation data obtained in the course of multiple analyses with NESSUS/FEM and NESSUS/FPI. Earlier versions of NESSUS fell short of utilizing the full extent of capabilities provided for in the database design. The development of new features for database access and data management in recent versions of NESSUS/FEM effectively opened up the use of the database to perform more sophisticated types of analysis.

The perturbation database resides in a binary (unformatted) direct-access file, and is structured as a two-way ordered linked list. This type of data structure allows the insertion, deletion and replacement of individual entries without the need to move large blocks of data. It is, therefore, possible to maintain and expand an existing database with results obtained in multiple runs of NESSUS/FEM. These capabilities are accessed with the use of the RECORD option, allowing the user to add or replace individual perturbed solution sets. This option makes efficient use of the data structures already implemented in the perturbation database, and performs updates by relocating points within the linked list.

An example of the data manipulations performed with the RECORD option is schematically depicted in Figure 2.4. An initial run of NESSUS/FEM is performed for three increments of static analysis with two perturbations on each of the first two increments. If all data sets are recorded, the resulting perturbation database will look as shown on the left in the figure. Further investigation of these might indicate the lack of a satisfactory result (for instance, lack of a converged solution) for perturbation 1 of increment 0. In addition, it becomes apparent that three additional random variables should have been included in the analysis for increment 0. Hence, a second run of NESSUS/FEM is performed for increment 0, recording only the new values for perturbation 1 (thereby superseding the earlier results) and perturbations 3 through 5 (corresponding to a perturbation of each of the three added random variables). Any computations not needed for the calculation of the modified or added perturbations can be skipped on the second run. The perturbation database, updated after the second run, will be structured as shown to the right in Figure 2.4. This degree of flexibility allows very efficient use of the perturbation database

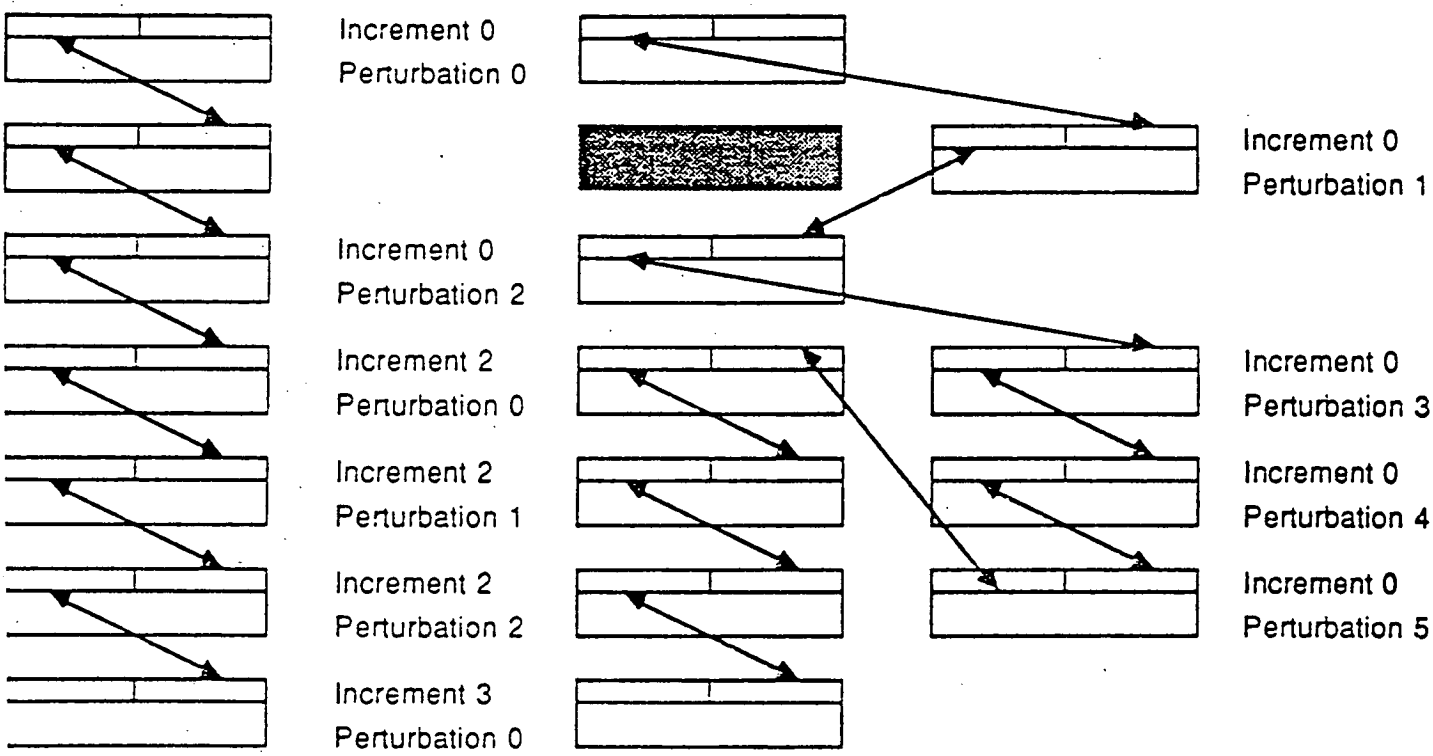


Figure 2.4 The Perturbation Database is Used to Maintain a Permanent Record of the Analysis History for a Given Model

for realistic problems which may require several test runs to obtain a good local representation of the response sensitivity.

For problems in which preliminary results indicate that the limit state lies well beyond the range of the perturbations used to determine the response sensitivity, it would be desirable to provide a way to reformulate the perturbation problem at a point closer to the limit state. This would allow the computation of accurate point probability estimates in the tails of the distribution, even though the response characteristics at the tails may be considerably different from what is observed near the means. This is possible with the use of the MOVE option, which redefines a new deterministic (unperturbed) state at a point other than the mean (see Figure 2.5).

In a way, the MOVE option represents the probabilistic counterpart of a well-known deterministic design practice. An experienced engineer will often choose to base his design on an analysis involving an extreme loading combination (worst loading case) acting upon a weak structure (with nominal material properties somewhat below the mean values). Using a reliability-type formulation, the location of the "design point" will provide the most likely combination of random variables that will result in the limit state being exceeded. Based on this information, it is possible to use the MOVE option to manipulate the random variables in order to reproduce the structure most likely to exceed a given limit state.

A print-out of the new unperturbed problem at the redefined deterministic state is included in the output from NESSUS/FEM. This provides a convenient way of checking for input errors in random variable definitions. In addition, since a complete resolution of the problem is performed at the redefined deterministic state, the MOVE option also provides a (somewhat expensive) way of checking the results obtained with the perturbation algorithms.

As indicated above, the perturbation database resides on an unformatted (binary) direct-access file which cannot be displayed or edited with a text editor. However, for small problems, it would be desirable to be able to generate a formatted translation which could be displayed at a console or sent to a line printer. This can be done by using the DECODE utility provided with NESSUS (see Figure 2.6). This utility is a stand-alone program which provides a formatted translation of a binary perturbation database generated by NESSUS/FEM. The original binary database file can be regenerated from its formatted translation by using the ENCODE utility, also provided with

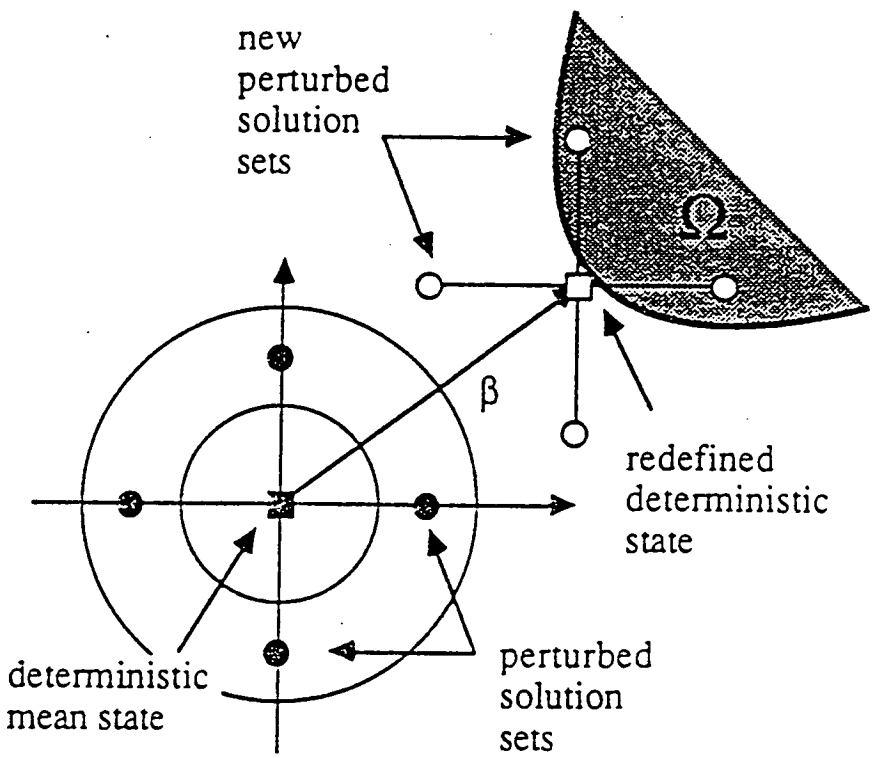


Figure 2.5 Improved Reliability Estimates are Obtainable by Redefining the Deterministic State at a Point Away from the Mean

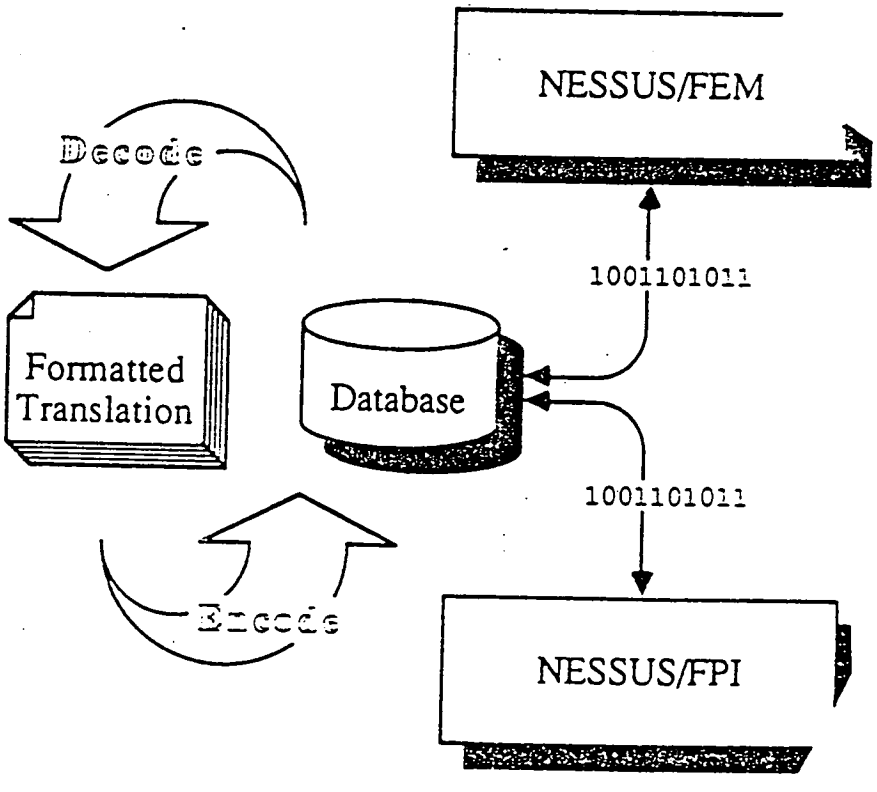


Figure 2.6 DECODE and ENCODE Utilities Enhance the Portability of the Perturbation Database

NESSUS. A small loss of accuracy is incurred in the process, since the formatted translation only carries five digit accuracy in order to fit the data from the largest record within one 132 character line. In spite of the shorter precision, the formatted translation files still occupy more memory than the (more compact) binary files, so that it may not be practical to obtain formatted translations of very large database files.

Unlike the binary direct-access files, which have machine dependent format, the formatted translation can easily be ported between different computers and operating systems. Hence, the availability of ENCODE and DECODE utilities on different machines allows the exchange of database files generated by NESSUS/FEM. With the emergence of smarter network software, such as NFS, the need to physically move database files between computers may no longer be as important. However, until these smarter networks come into widespread use, the formatted database translation will continue to provide a standard format for the exchange of database files.

2.7 Other Enhancements and Improvements

Several other enhancements and improvements were introduced in NESSUS/FEM in the course of the past years. These enhancements reflect needs that were identified by exercising NESSUS on a variety of realistic engineering problems and across a broad spectrum of computing equipment and operating systems. What follows is merely a list of some of the most visible enhancements implemented in this period.

Early attempts to use existing finite element meshes for probabilistic analysis with NESSUS/FEM identified the need to allow for collapsed configurations of the standard continuum elements. This raised some conflicts with the nodal strain projection algorithm used in NESSUS, since the use of nodal quadrature requires a well-defined Jacobian for the isoparametric mapping at each node. This problem was avoided by allowing the use of row-sum lumping to form the "lumped volume" matrix used in the strain projection algorithm. Although the use of row-sum lumping is not as accurate as with nodal quadrature, this strategy allowed the degeneration of continuum elements to form triangles, wedges and tetrahedra in order to preserve the topology of existing meshes. Due to its superior performance, the use of nodal quadrature is still recommended for most regular meshes without collapsed elements. additional efforts to improve the performance of row-sum lumping are planned for the coming year.

The existing format for the input of anisotropic material properties was found to be inconvenient for the input of simple types of anisotropic materials which are being investigated for use in SSME components. The materials in question are single-crystal alloys exhibiting cubic symmetry (such as PW1480) and amenable to a three-parameter material description. Furthermore, it was desirable to allow all three parameters to be random, which could not easily be done using the input format for general anisotropy. As a result, a special extension to the isotropic material properties input reader was implemented to allow the specification of a three random parameter material model. This feature has been used extensively in the analysis of an SSME HPFTP turbine blade model at Rocketdyne.

During the eigenvalue analysis of some structural problems using subspace iteration, the matrices on the reduce eigenproblem were found to differ by several orders of magnitude, resulting in a very poorly conditioned problem. This often resulted in overflow problems during Jacobi iteration on machines that use a large mantissa with small exponent (such as the D-float format on VAX). The problem was cured by using a spectral transformation to improve conditioning of the problem in the subspace. A better algorithm for selecting the trail vectors also helped improve the performance of the algorithm.

The eigenvalue perturbation algorithm currently implemented in NESSUS was subjected to a very extensive cleanup in order to remove a number of existing bugs, streamline the code and improve its reliability. In addition, the convergence criteria used to stop the recursive algorithm have been modified significantly. If an elastostatic analysis is used to obtain the initial stresses prior to the dynamic mode extraction, it is now possible to change the convergence criteria for the eigenvalue problem from the values used in the static perturbation analysis.

A new option to extract the deformation modes present in the assemble stiffness matrix is available in NESSUS/FEM. This option involves the solution of the standard eigenvalue problem.

$$(K - \lambda I)\, x = 0 \qquad (2.5)$$

The resulting eigenvalues represent allowable deformation modes for the assemble stiffness, and the corresponding eigenvalues indicate the strain energy associated with the mode. This information is very useful for the

development of new element formulations and to obtain stability estimates for problems involving perturbations to the stiffness matrix.

A new MONITOR facility was introduced to provide a convenient way to monitor the behavior of critical response variables in the course of iteration. A summary of the current values for all monitored variables is printed on the log files at every iteration. During interactive execution, the log file is displayed on the terminal screen, allowing the user to track these quantities while the iteration is in progress.

Until recently, the transient dynamics capability using direct integration of the (deterministic) equations of motion was not active in the NESSUS code. Following a major cleanup of this analysis driver performed under the auspices of the MHOST project, this option has been reactivated and tested in NESSUS/FEM.

A single-step direct time integration scheme based on the Newmark-β family of algorithms is used. Individual schemes within this family of algorithms may be obtained by selecting the control parameters for the Newark algorithm as follows:

γ	β	INTEGRATION SCHEME
1/2	0	Central Explicit
3/2	1	Backward Difference
1/2	1/10	Linear Acceleration
3/2	4/5	Galerkin
1/2	1/12	Fox-Goodwin
1/2	1/4	Average Acceleration

The "average acceleration" scheme is the system default, with the stress and strain recovery at the end of each time step. Only Rayleigh-type damping may be used in this type of analysis. In addition to all the mechanical loadings available for static analysis, a general periodic loading or displacement constraint can be used, with period and amplitude both specified on a nodal basis. Nodal displacements, velocity or acceleration may be specified as part of the initial conditions for the dynamic problem.

Finally, the MENTAT compatible post-file writer in NESSUS was extended to include mode shapes for vibration, buckling and deformation mode analysis. The output for eigenvalues and eigenvectors follows the former MARC K.1 post-file format and is fully supported by the current commercial version of MENTAT. As with the static problem, only the unperturbed eigenvalue solution is written to the post-file.

2.8 Random Vibration for Uncertain Structures

The code used to perform random vibration analysis (PSD) in NESSUS was the object of extensive clean-up as a first step towards extending the current capabilities to include uncertain structures as well. A strategy for the implementation of random vibration analysis of uncertain structures is currently being laid-out. The proposed implementation is based on the use of the approximate natural frequencies and mode shapes for the perturbed structure, obtained with the eigenvalue perturbation algorithm, to provide information on the sensitivity of the RMS stress and displacement to small fluctuations of the random variables.

A more sophisticated capability will involve the introduction of the PSD level itself as a random variable. The PSD level will have to be handled as a special type of random variable since it is irrelevant to the perturbed eigenvalue computation and will only affect the computations carried out in the frequency domain. The introduction of uncertainty in the PSD level may provide a systematic alternative to the more conservative practice of constructing an envelope to the PDS function.

2.9 Future Effort: Nonlinear and Transient Problems

The iterative perturbation approach adopted in NESSUS/FEM appears suitable for extending the existing formulation to situations involving at least mild nonlinearities. The basic solution strategy will amount to tracking multiple perturbed time-histories, using the solution to the unperturbed problem as the iteration preconditioner for all perturbed problems at a given time step or load increment. Difficulties will arise if some of the perturbed problems drift too far from the unperturbed state in the course of an analysis. The problem may be aggravated by the presence of constraint equations, which arise naturally from the formulation for deviatoric rate-independent plasticity.

Other problems are expected in situations involving repeated cyclic loading, since the response for intermediate perturbations is not necessarily bounded at all times by the response corresponding to the largest perturbations. This problem has been observed in works by other researchers dealing with transient dynamics, in which the variance of the response appears to vanish at several points in time [16,17]. No solution has been offered for this problem.

Yet another problem involves the emergence of secular terms in the response for the perturbed system, which may grow unbounded in time and invalidate the solution for large times. This pathology is well known to researchers working on nonlinear oscillations of complex dynamical systems and there is extensive literature on the subject. This problem has been discussed by Liu and Belytschko [17], and appropriate secularity filtering strategies have been suggested.

Perhaps the most intractable problem in probabilistic nonlinear mechanics involves the presence of bifurcations, in which very small perturbations of the deterministic problem can lead down very different solution paths. With present finite element technology, these problems can become extremely complex even for deterministic analysis. However, the problem of detecting the presence of a nearby bifurcation point represents a much simple problem, involving the solution of a stochastic eigenvalue problem.

References

[1] S. Nakazawa, J.C. Nagtegaal and O.C. Zienkiewicz, "Iterative Methods for Mixed Finite Element Equations." AMD Vol. 74, ASME, 57-67 (1985).

[2] S. Nakazawa, "Mixed Finite Elements and Iterative Solution Procedures," Innovative Methods in Nonlinear Problems, Chap. 21, (W.K. Liu, T. Belytschko and K.C. Park, eds.), Pineride Press, Swansea (1984).

[3] S. Ahmad, B.M. Irons and O.C. Zienkiewicz, "Curved Thick Shell and Membrane Elements with Particular Reference to Axisymmetric Problems," Proc. 2nd Conf. Matrix Methods in Structural Mechanics, Wright-Patterson A.F. Base, Ohio, 1968.

[4] T.J.R. Hughes and T.E. Tezduyar, "Finite Elements Based Upon Mindlin Plate Theory with Particular Reference to the Four-node Bilinear Isoparametric Element, J. Appl. Mech., 48, 587-596 (1981).

[5] H.C. Huang and E. Hinton, "A Nine Node Lagrangian Mindlin Plate Element with Enhanced Shear Interpolation," Eng. Comput., 1, 369-379 (1984).

[6] W.P. Doherty, E.L. Wilson and R.L. Taylor, "Stress Analysis of Axisymmetric Solids Utilizing Higher Order Quadrilateral Finite Elements," SESM Report 69-3, Dept. Civil Engineering, University of California, Berkeley (1969).

[7] K. Kavanagh and S.W. Key, "A Note on Selective Reduced Integration Techniques in the Finite Element Method, "Int. J. Num. Meth. Engng., Vol. 4, No. 1, 148-150 1972).

[8] D. Kosloff and G.A. Frazier, "Treatment of Hourglass Patterns in Low Order Finite Element Codes," Num. Anal. Meth. in Geomech., 2, 57-72 (1978).

[9] D.P. Flanagan and T. Belytschko, "A Uniform Strain Hexahedron and Quadrilateral with Orthogonal Hourglass Control," Int. J. Num. Metho. Engng., Vol. 17, 679-706 (1981).

[10] E.L. Wilson, R.L. Taylor, W.P. Doherty and J. Ghaboussi, "Incompatible Displacement Models," Numerical and Computers Methods in Structural Mechanics, (S.J. Fenves, N. Perrone, A.R. Robinson and W.C. Schnobrich, eds.), Academic Press (1973).

[11] R.L. Taylor, P.J. Beresford and E.L. Wilson, "A Nonconforming Element for Stress Analysis," Int. J. Num. Meth. Engng., Vol. 10, No. 6, 1211-1219 (1976).

[12] T.H.H. Pian and K. Sumihara, "Rational Approach for Assumed Stress Finite Elements, "Int. J. Num. Meth. Engng., Vol. 20, 1685-1695 (1984).

[13] T. Belytschko and T.J.R. Hughes, Computational Methds for Transient Analysis, North-Holland (1983).

[14] A.E. Green and P.M. Naghdi, "A General Theory of an Elastic-Plastic Continuum, "Archive for Rational Mechanics and Analysis," 18, 251-281 (1965).

[15] J.C. Nagtegaal and J.E. deJong, "Some Aspects of Nonisotropic Workhardening in Finite Deformation Plasticity," Proc. Workshop on Plasticity of Metal at Finite Strain: Theory, Computation and Experiment, (E.H. Lee and R.L. Mallett, eds.), Division of Applied Mechanics, Standford University (1981).

[16] S. Nakagiri, T. Hisada and K. Toshimitsu, "Stochastic Time-History Analysis of Structural Vibration with Uncertain Damping," PVPVol. 93, ASME, 109-120 (1984).

[17] W.K. Liu, T. Belytschko and G. Besterfield, "Transient Probabilistic Finite Element Systems," Proc. WCCM '86, Austin, Texas (1986).

3.0 NESSUS PROBABILITY ALGORITHM DEVELOPMENT

3.1 Introduction

This chapter summarizes the probability algorithms developed for the NESSUS code. Two methods of probability modeling are to be included. The first of these is the Fast Probability Integration (FPI) method [1,2]. The second method is the Monte Carlo method. Both methods use the same structural sensitivity data, which is generated by NESSUS. Confidence levels will be estimated for the response variables distributions that are calculated.

The development of the Monte Carlo methods, performed at the University of Arizona, is completed. A summary of the Monte Carlo methods is included in Section 3.2. Among the four methods investigated, the Harbitz method is considered the best method, therefore, it will be integrated into the NESSUS code.

Section 3.3 describes the on-going development of the method for estimating the probabilistic solution for the entire structure using limited perturbation solutions at selected nodes.

Section 3.4 discusses a strategy for integrating the FPI and the Monte Carlo codes. The issue of defining the proper perturbation ranges is addressed.

Section 3.5 defines a code-structure that extends the NESSUS/PRE capability from normal to non-normal correlated random variables. The development of the FORTRAN routines for performing the variable transformations is complete. In the future, minor modifications of the PRE and FEM modules will be required to integrate the codes.

Section 3.6 describes two enhanced FPI iteration algorithms. One algorithm is for solving a response value given a specified probability value. The other algorithm is for solving a probability value given a specified response value. As demonstrated by an example, these algorithms provide very efficient solutions.

Section 3.7 demonstrates, using one of the validation problems, the confidence bound estimation procedure. The procedure is consistent with the NESSUS/FPI solution algorithms.

3.2 Fast Monte Carlo Methods

Consider the random variable Z as a function of the random vector

$$\underset{\sim}{X} = (X_1, X_2, .. X_n)$$

$$Z = h(\underset{\sim}{X}) \tag{3.1}$$

The distribution of each X_i is known. It is assumed that all X_i are mutually independent.

A fundamental problem of probabilistic mechanics and design is to compute a point probability,

$$p = P[h(\underset{\sim}{X}) \leq h_o] \tag{3.2}$$

For example, p could represent the probability of exceedance of a deflection or perhaps the probability of failure.

Another problem is the extension of the first to the construction of a cumulative distribution function.

$$F_Z(z) = P[h(\underset{\sim}{X}) \leq z] \tag{3.3}$$

Clearly, the two problems are identical, but optimal strategies for analysis may differ. For example, to construct the CDF, one option would be to obtain point estimates of F_Z at selected values of z, then fit a curve through the points. A second option would be to construct an empirical distribution function from a large sample of Z_i. There are a number of Monte Carlo techniques which can be employed to estimate p and/or F_Z.

Monte Carlo traditionally has been considered to be a "last resort" method for solving a probability or statistics problem because of high cost relative to accuracy of the results. However, in recent times a combination of the development of new efficient numerical techniques and new digital computing hardware have made Monte Carlo more attractive.

Appendix C presents descriptions of the following Monte Carlo programs dedicated to probabilistic structural analysis.

1. **"Conventional" Monte Carlo:** For conventional Monte Carlo, a random sample of $\underset{\sim}{X}$ is obtained. In turn, a random sample of Z is computed using Eq. (3.1). An empirical distribution function of Z is constructed.

2. **Variance Reduction Using Antithetic Variates:** Given a sample of $\underset{\sim}{X}$, a negatively correlated "mirror image" $\underset{\sim}{X}'$ is computed. The variance of point probability estimates is reduced by averaging the estimates made by $\underset{\sim}{X}$ and $\underset{\sim}{X}'$.

3. **Mean Value Method with Stratified Sampling:** This method directly evaluates a multiple integral expression for point probabilities.

4. **The Harbitz Method:** This is a scheme for reducing the sample space for $\underset{\sim}{X}$ thereby, in theory, producing efficient point probability estimates.

Results of the performance study are summarized in Figure 3.1 where CYBER 175 CPU time is plotted as a function of probability level β and number of variables, n. It is important to note that β is related to the tail probability level p by

$$p = \Phi(-\beta) \tag{3.4}$$

where Φ is the standard normal CDF. Computer time for each method depends on factors other than probability level and number of variables. The distribution type for each factor and the form of the response function influence computation time. Therefore, the curves of Figure 3.1 must be interpreted as characterizing the relationships for purposes of comparison.

Several general conclusions can be made regarding the results presented in Figure 3.1.

1. Fast probability integration (e.g., the Wu/FPI method) is far more efficient than Monte Carlo.

2. Variance reduction does not appear to be competitive with the other methods.

3. For small numbers of variables, the mean value and Harbitz methods are very efficient with the Harbitz method having a slight edge.

4. Computing time for both the mean value and Harbitz methods increases sharply as the number of variables increases.

5. For small numbers of variables, conventional Monte Carlo is not efficient. But the increase in computing time increases linearly with the number of variables. Because these curves are flatter than the mean

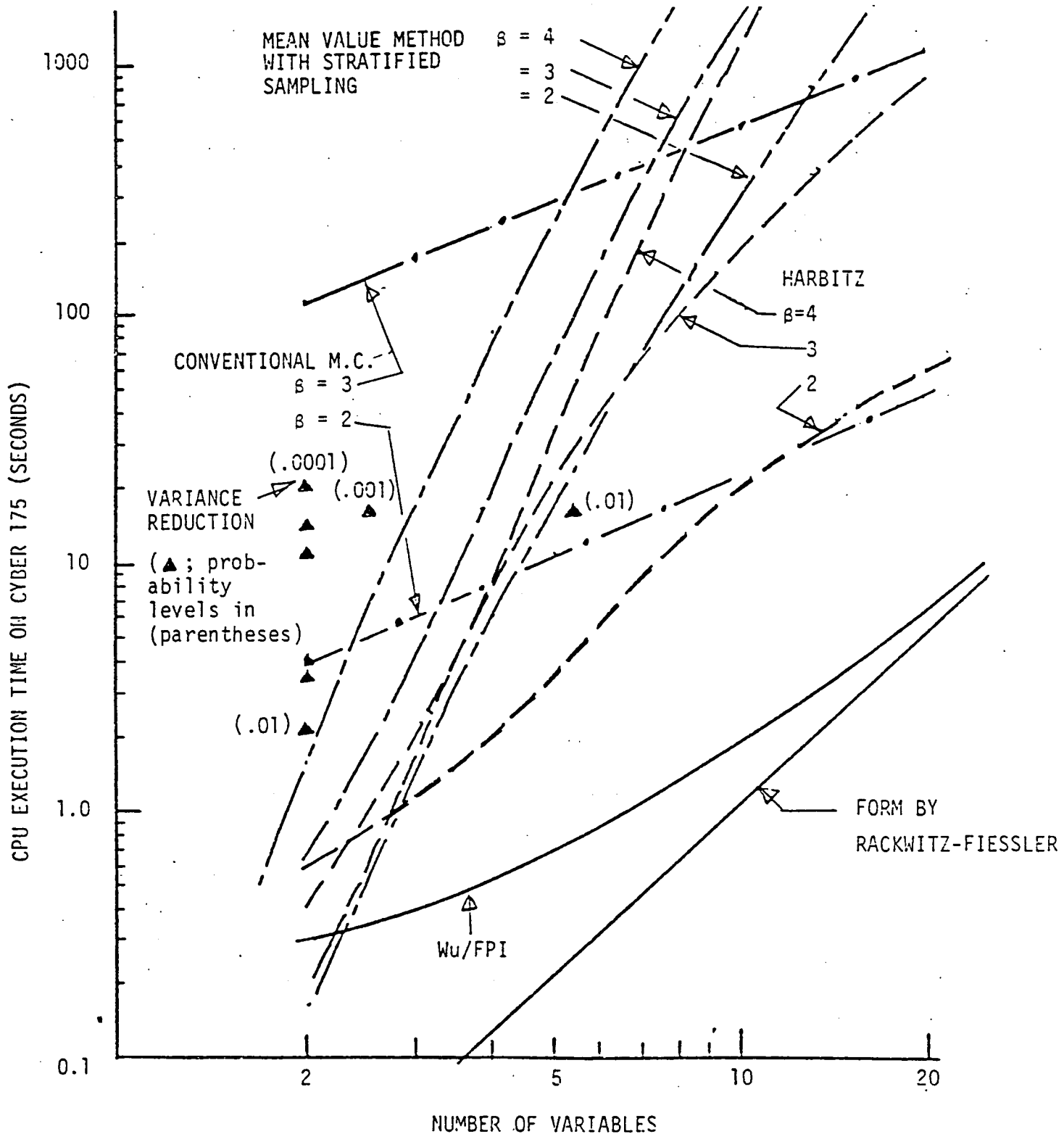


Figure 3.1 A Summary of Efficiencies of Four Monte Carlo Methods for Computing Point Probabilities (for Monte Carlo ±10% Accuracy with a Confidence of 95%)

value or Harbitz curves, conventional Monte Carlo actually becomes more efficient relative to each of these methods above a given n.

6. Conventional Monte Carlo gets very expensive as the probability level decreases. Note that the $\beta = 4$ curve is off of the chart.

7. One feature of conventional Monte Carlo is that a full sample of the response variable can be generated. Therefore, the entire CDF of the response variable can be generated. On the other hand, several probability points have to be computed using the other methods. And the accuracy will be better for larger probability levels and worse for smaller p.

In summary, a general conclusion is that the Harbitz method seeems to be the preferred approach. Note, however, as the probability level p gets larger (and β smaller), the Harbitz method approaches conventional Monte Carlo.

3.3 The Probabilistic Field Problem

Probabilistic structural analysis using the NESSUS code requires constructing response function surface for each response variable. Such response surfaces can be constructed using curve fitting schemes. The NESSUS probabilistic solution strategy is to use only low-degree (i.e., first- and second-degree) polynomial surfaces because higher-degree surfaces are difficult and impractical to construct using the NESSUS generated response solutions. For such low-degree surfaces to be useful for generating accurate probability information, it is necessary to make a good selection of the response solution points for response surface approximations.

In the current NESSUS technology, the selection of the solution points or regions is based on the "most probable point" (or design point) concept [2-4]. The validation studies (see Section 6.0) indicate that the above strategy works well. However, for the solution to be accurate, the method requires, in addition to the mean-based perturbation, the deterministic re-computation/correction of the response value at the most probable points. For further improvements, more perturbations may be required around the most probable points (see Section 3.6). In general, these most probable points are different for each response variable in the structure. For example, the most probable points for the stress at node 1 may not be the same as for the stress at node 10.

In probabilistic structural analysis, it may be necessary to generate probability-based solutions for the entire structure under analysis. One reason is that the sub-critical areas identified from the conventional

deterministic solution may become critical, from the probabilistic point of view, if the areas are subjected to more uncertainties in loads, material properties, geometries, etc.).

A useful probabilistic solution, in contrast to the deterministic solution, is the responses (e.g., stresses) at all nodes at selected probability of exceedance levels. To generate the field solutions, it may be extremely time-consuming to perform "full" NESSUS probabilistic analysis for "every" response variable because each response variable requires its own perturbation. Therefore, it is important to develop a strategy to obtain approximate probabilistic response field without having to solve each response variable independently.

To solve the above field problem, work has been initiated to formulate an estimation strategy based on the most-probable-point-locus concept [4]. A preliminary solution for the field problem will be discussed in the following paragraphs. More detailed study of the field problem is in progress. A computer program has been written to study and test several strategies. The goal is to investigate strategies and make recommendations for the code implementation.

As a first approximation, the field's response can be made using the mean value first order (MVFO) database at the mean solution. This technique may be used to identify regions of greatest concern (high probability of exceedance) in the structure. However, high accuracy for the probability of exceedance throughout the entire field cannot be obtained for highly-nonlinear response surfaces using only the MVFO database.

However, if the response variables are statistically correlated within the regions of concern, it may be possible to predict or estimate the regions' field response based on a small number of accurate solutions for the "critical" response variables.

A sample demonstration was selected to study the response field problem. The example consisted of a "fix-free" bar is subjected to an axial force. The bar has two elements with Young's modulus E_1 and E_2, respectively. If we assume that E_1 and E_2 are mutually independent random variables, the longitudinal deflections constitute a response field that is nonlinear in the random variables. Based on a detailed FPI study of this example, the following preliminary conclusions were reached:

(a) The correlation between any two response variables, measured by the correlation coefficient (ranging from - 1 to + 1), can be estimated using the mean-value solutions.

(b) Reasonably good probabilistic solutions can be predicted from the solution of one response variable to the other, provided that the response variables are reasonably well-correlated (e.g., correlation coefficient > 0.7 or < -0.7).

(c) The quality of the estimates depend on which response variable is used as a reference or "master" variable. This master response will provide the common computation points for the computations of the master as well as the other "slave" response variables. Because the selected points are the most probable points for the master, naturally the master response has the highest accuracy. The accuracies of the slaves depends on the correlation coefficients. In general, the accuracy will decrease as the magnitude of the correlation coefficient becomes smaller. This suggests that it is important to select a good reference point. In general, a master may be selected, based on the MVFO solution, as the critical response (e.g. maximum stress) at a selected probability level.

(d) When the correlation coefficients become far from unity (plus or minus) between a master and a slave, then a new reference point may be required. In general, several reference points may be selected after the mean value perturbations.

3.4 The Integrated NESSUS/FPI/Monte Carlo Algorithm

In the NESSUS analysis, the FPI algorithm is being applied at two levels. At the first level, the NESSUS/FPI code generates probabilistic output using the established response function established based on the NESSUS database. At this level, NESSUS/FPI is accurate relative to the accuracy of the response function. At the second level, which is most critical to the NESSUS accuracy, the FPI algorithm directs the FEM module to "move" to other perturbation centers (the most probable points generated from NESSUS/FPI). The first level is always efficient because the response function is explicitly defined. At the second level, however, finite element solutions are required to define the response function (i.e., the response function is implicitly defined), and the computation time becomes dominant.

The NESSUS Monte Carlo algorithm is applied as an alternative to the NESSUS/FPI only at the first level. The major reason is, based on the result of the studies of the Monte Carlo methods (Section 3.2), it appears that it is practically impossible to perform Monte Carlo simulation by actually generating a "sufficient" number (e.g., thousands or more) of FEM solutions.

The advantage of including a Monte Carlo module is that Monte Carlo simulation has the capability of providing exact solutions (as the number of

samples becomes larger) and involves less potential numerical/convergence problems than the NESSUS/FPI algorithm. Therefore, the Monte Carlo module can be used for independently checking the NESSUS/FPI results.

It is planned that the Monte Carlo module will be independent in the NESSUS system and that this module will be controlled by the PFEM module. The users will have the options of selecting the Monte Carlo or the NESSUS/FPI solution type.

Because Monte Carlo simulation will not be applied to generate the FEM solutions, the accuracy of the NESSUS will rely on the FPI algorithm (applied at the second level). To avoid gross error, a strategy is described in the following paragraphs which suggest that "large" perturbation solutions can be generated to fit a response surface.

In applying the FPI algorithm, there is a possibility that the established (up to second-degree polynomials) response surface do not represent very well the actual response surface. Originally, the FPI algorithm required only good fit of the response surface in the neighborhood of the most probable point. In other words, only "small" perturbations are required. However, it is not impossible that the response surface may require higher than a second-degree model for its accurate description, or that more than one local most probable point exists for the surface. Please note that this is based on theoretical considerations. It has been demonstrated that FPI provides high accuracy for all the validation problems performed, even with only linear surface approximations.

To provide the analysts with more confidence, "large" perturbation solutions can be generated so that the solution points cover a "wide" range. If there are no significant differences in the solutions, then there is more confidence that the solution, based on the lower-order response surface, is correct. If the results show significant differences, indicating that the response surface cannot be modeled adequately by a second-degree surface, then a more detailed analysis must be considered. A possible solution is to generate a higher-degree response surface and then use the Monte Carlo program. Note that after the regular NESSUS analysis, some probability information is already available; therefore, the higher-order effect needs to be considered only for those significant random variables.

3.5 Non-normal Random Variables - NESSUS/PRE

The NESSUS/PRE module was originally designed to solve problems involving statistically correlated normal random variables. The PRE module generates a transformation matrix, [T], using the covariances of the correlated variables, such that

$$[Y] = [T][Z] \tag{3.5}$$

where

[Y] = a statistically correlated normal vector, and

[Z] = a vector of un-correlated normal

The distributional input data requires only mean and standard deviation. The output of the NESSUS/PRE code includes the [T] matrix, which is required for the NESSUS/FEM input data. The PRE module has been tested successfully in a number of validation problems (see Section 6.0).

For a correlated normal vector of random variables, the NESSUS solution procedure is straight-forward mainly because PRE is a totally independent module. The extension of the correlated normal model to the correlated non-normal model is based on a methodology developed for the PSAM project [2]. The procedure is more involved and requires additional input and subroutines in the PRE and FEM modules. During the last year, several strategies, including the use of the NESSUS/EXPERT, have been investigated. The final structure has been defined and will be implemented in the next year code.

Let [X] be a vector of correlated, non-normal variables. The input of the PRE module will be modified to include several distributional types (lognormal, Weibull, etc.). A subroutine will be added to the PRE module to transform [X] to [Y] using a transformation [2] abbreviated as

$$x = f(y) \tag{3.6}$$

where x and y are the elements of [X] and [Y], respectively. An "equivalent" covariance matrix for [Y] will be generated and then used to generate [T].

In the FEM module, for a perturbation in z (element of [Z]), [Y] is computed using (3.5). An additional subroutine will be added to transform Y to X using (3.6).

3.6 NESSUS Probabilistic Solution Iteration Algorithms

The basic probabilistic analysis algorithm for the NESSUS has been developed [2] and validated using a number of problems (Section 6.0). For constructing the entire cumulative distribution function (CDF), the algorithm has proven to be effective. However, the current procedure is not satisfactory if the analysts need only one or a few points on the CDF curve.

To optimize the iteration procedure, two algorithms, one for specified probability levels, and the other for specified response levels, have been formulated to be used in the PFEM module. The first algorithm (for user-specified probability level) is illustrated in Figure 3.2 using validation Case 3 (see Section 6.0 - beam natural frequency). The procedure is as follows:

(a) Select a probability level.

(b) Compute the most probable point using the MVFO method.

(c) Recompute the response at the most probable point. (Note: the solution is called the advanced MVFO, or AMVFO solution)

(d) Conduct NESSUS perturbation around the most probable point. (Iteration around the most probable point)

(e) Go to (b) and repeat the process until response value converges.

To implement the above procedure, the NESSUS/FPI code has been modified to solve the above step (b) for any user-specified probability level. The entire solution requires the PFEM module to interface the FPI and the FEM modules. It is expected that the solution should converge in a fast rate. In the present example, an accurate solution is obtained with the AMVFO method, i.e., no iteration is required.

The second algorithm (for user-specified response level) is illustrated in Figure 3.3 using the same validation case. The procedure is somewhat complicated but is highly efficient in minimizing the NESSUS/FEM computations. The procedure is as follows:

(a) Using the MVFO method, construct CDF curve using the NESSUS/FPI code to get the intercept and the slope at the 50% probability level.

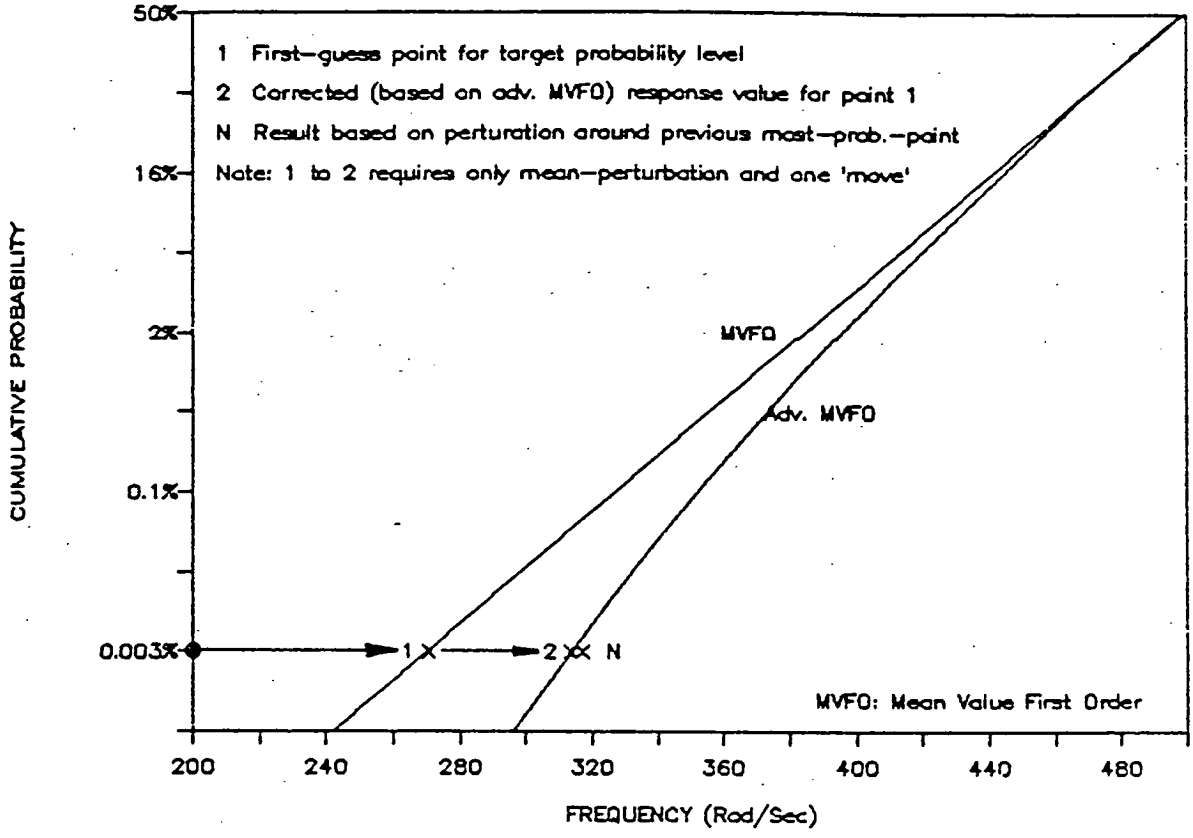


Figure 3.2 Iteration Algorithm A for Specified Probability Level

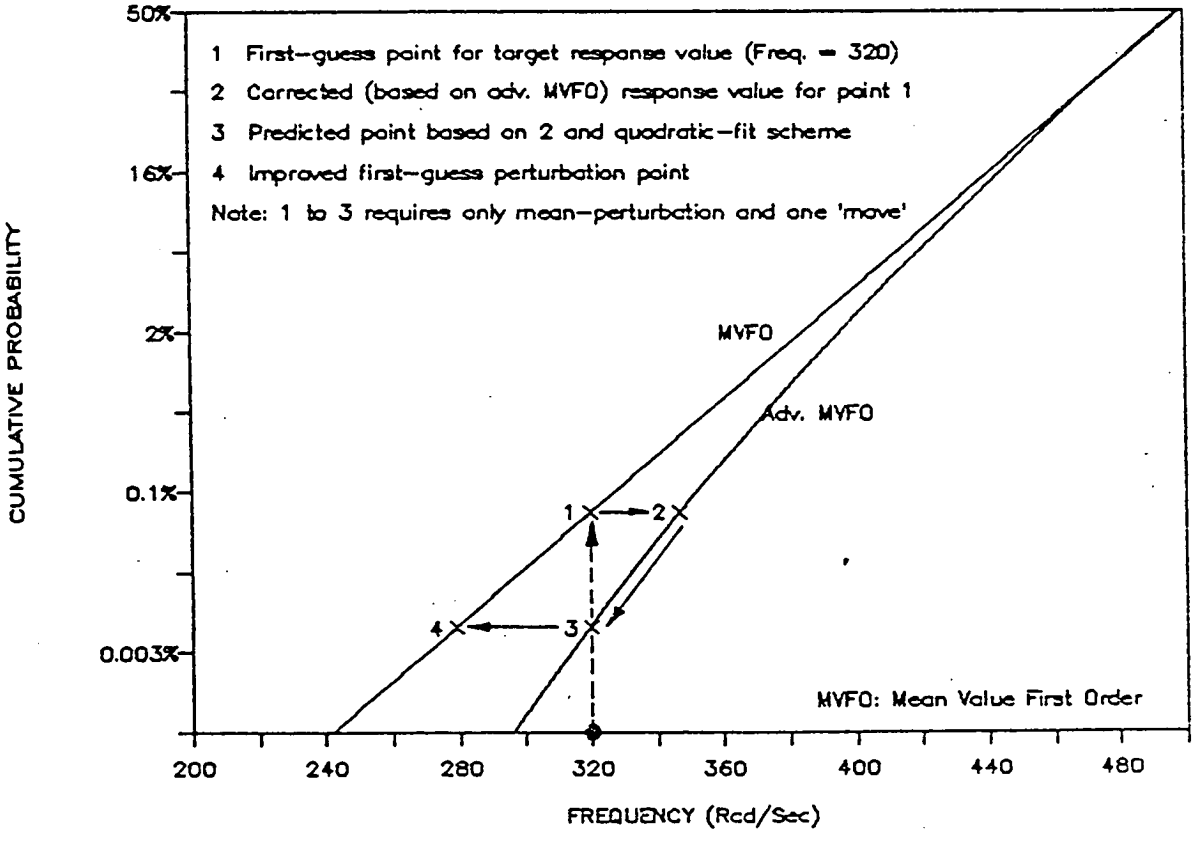


Figure 3.3 Iteration Algorithm B for Specified Response Value

(b) Select a response value (e.g., frequency = 320 Hz).

(c) Compute the corresponding point on the MVFO solution curve (i.e., point 1 in Figure 3.3) and compute the most probable point using the MVFO method.

(d) Recompute the response at the most probable point (i.e., locate point 2 in Figure 3.3).

(e) Use point 2 and the information from step (a) to fit a quadratic curve. Use this curve to predict the probability level at the specified response level (i.e., locate point 3 in Figure 3.3)

(f) Compute the corresponding MVFO response for point 3 (i.e., find point 4 in Figure 3.3).

(g) Compute the most probable point at point 4 and use this point as a starting point for iteration.

(h) Start iterations about target response value. Iteration stops when the probability level converges.

The implementation of the above procedure requires the use of the PFEM module to integrate FPI and FEM modules. Because of the quadratic curve fitting scheme, it is expected that the solution should converge quickly. In the present example, the curve-fitting solution point (point 3) falls almost exactly on the AMVFO curve indicating the effectiveness of the quadratic fit. Note that point 3 in Figure 3.3 requires only mean-perturbation and an additional FEM deterministic solution.

3.7 Confidence (Error) Bounds Estimation

The NESSUS probability estimation algorithm described in Section 3.5 has assumed that the statistical distributions of the random variables are known. When the distributions are not certain because of the limited samples, the PSAM approach is to model the distribution parameters (mean, m, and standard deviation, s) as random variables, and then establish the distribution of the response CDF for specified response values. [2]

Consider an input random variable X. m and s are modeled as normally distributed and lognormally distributed variables, respectively. Given a sample with size n, the COVs (coefficient of variation = standard deviation/mean) for m and s are [2]:

$$C_m = C_x / \sqrt{n} \tag{3.7}$$

$$C_s = 1 / \sqrt{2(n-1)} \tag{3.8}$$

Where C_x is the COV of the input random variable X, C_m and C_s are the COVs of m and s, respectively.

The NESSUS/FPI code has an option to compute confidence bounds. The extra input are C_m and C_s for each X. The output are the upper and lower bounds that contain 90% and 95% of the probability. The method for computing the bounds is a combination of the FPI method and the Monte Carlo simulation. More specifically, the response CDF (now becomes a random variable) is computed using FPI method for every randomly generated m and s sets [2].

A validation problem was solved using validation case 5 - Rotating Beam First Modal Frequency (see Section 6.0). The COV data are listed in Table 3.1 where n = 20 was assumed for all five input variables. Figure 3.4 shows solutions at three frequencies using the AMVFO method.

Table 3.1

Data for Confidence Bounds Example
(n=20; simulation sample size = 5,000)

X	C_x	C_m	C_s
Young's Modulus	0.10	0.02236	0.1622
Length	0.05	0.01118	0.1622
Thickness	0.05	0.01118	0.1622
Width	0.05	0.01118	0.1622
Density	0.05	0.01118	0.1622

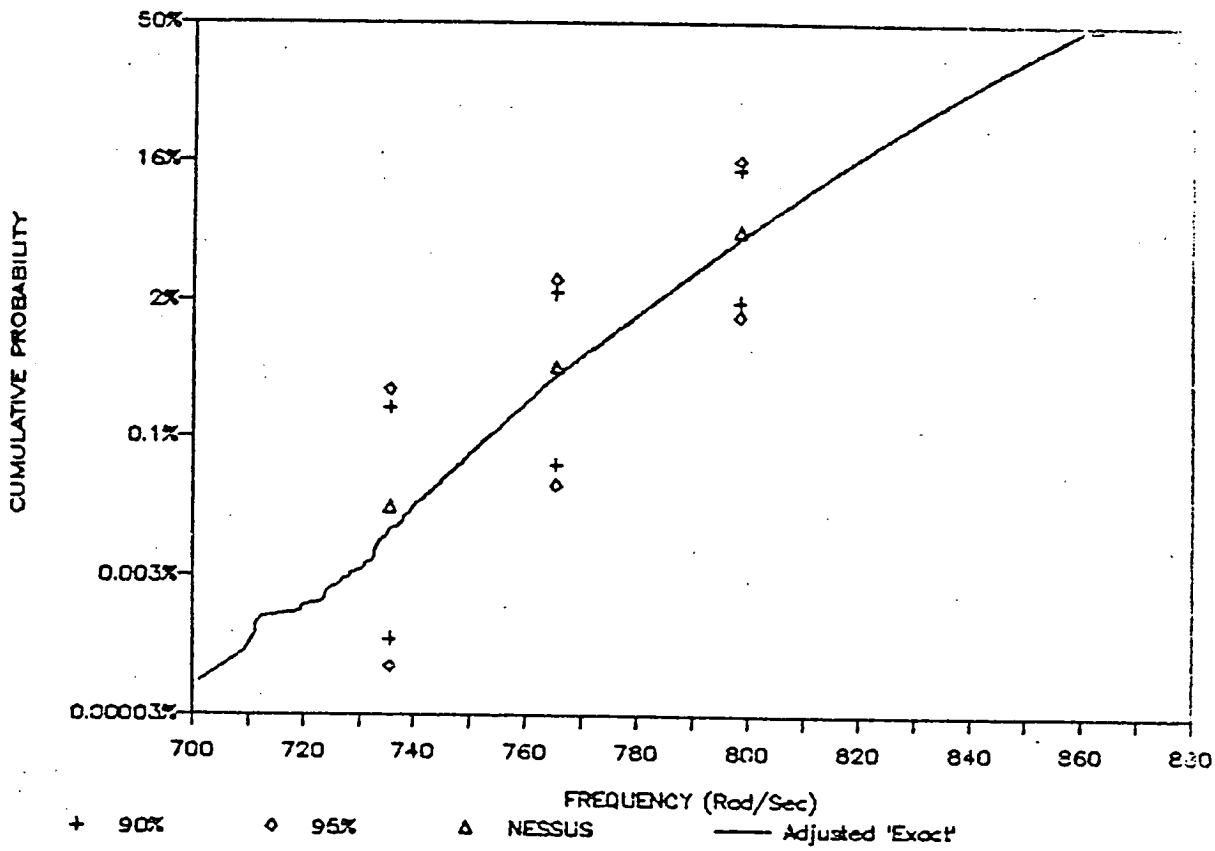


Figure 3.4 Example of NESSUS Confidence Bounds Estimation

References

[1] "Probabilistic Structural Analysis Methods (PSAM)," 1st Annual Report, NASA Contract NAS3-24389.

[2] "Probabilistic Structural Analysis Methods (PSAM)," 2nd Annual Report, NASA Contract NAS3-24389.

[3] T. A. Cruse, O.H. Burnside, Y.-T. Wu, E.Z. Polch and J.B. Dias, "Probabilistic Structural Analysis Methods for Select Space Propulsion System Structural Components (PSAM)," submitted to Computers & Structures, September 1987.

[4] Y.-T. Wu, O. H. Burnside and J. Dominguez, "Efficient Probabilistic Fracture Mechanics Analysis," Fourth International Conference on Numerical Methods in Fracture Mechanics, Texas, March 1987.

4.0 NESSUS/EXPERT SYSTEM CODE DEVELOPMENT

4.1. Summary

4.1.1. Change of Approach

As stated in last year's annual report, the type of knowledge that must be embodied in NESSUS/EXPERT fits, in a fairly straightforward manner, the production rule knowledge representation technique. This is convenient since most expert system building tools support this type of knowledge representation scheme. The main problem, at the start of this effort, was the lack of such tools that could integrate/communicate extensively with a system outside of its own environment. NESSUS/EXPERT requires integration with FORTRAN, so some time was spent searching for an expert system building tool written in FORTRAN. Consideration was even given to developing one for this project. However, due to the limitations of standard FORTRAN-77, especially the lack of recursion, the undertaking would not be trivial if a truly useful tool was to be developed. Thus, the tool called OPS5 was selected because of its ability to at least access the Lisp environment, and because it ran on a DEC VAX.

Near the end of the 1986 calendar year, DEC began to market a version of OPS5 written in Bliss that could access the non-Lisp environments on the VAX (including FORTRAN). Since that time, vendors have progressed towards offering some tools that can access non-Lisp environments, mainly because the tool is not written in Lisp, but a more conventional programming language - usually C. One such tool is CLIPS.

As a result of the emergence of such tools, some time at the beginning of the 1987 calendar year was spent analyzing the effects of changing tools in the middle of the project. A port from the public domain version of OPS5 to the DEC OPS5 was made so that the interface to FORTRAN could be assessed. At the same time, a re-assessment of NESSUS/EXPERT was made and its functionality was divided into areas that should use rule-based vs. FORTRAN-based methods of implementation. The division was based not only on required functionality, but also on efficiency issues with the result being that the rule-based portion would perform all of the higher-level decision making and consistency checking between keywords while FORTRAN would do all of the lower-level checking required on the parameters and data associated with a single keyword.

Such a division of capabilities required an extensive ability to pass information back and forth between the rule-based portion of the system and the FORTRAN code. This amount of integration was not handled easily or efficiently in DEC OPS5, so CLIPS was examined more closely.

At the same time, the flow of control and menu interface implemented in NESSUS/EXPERT were examined in detail to determine their validity and appropriateness. As a result of this analysis, it was decided that certain improvements could be made. The main improvement required some redesign of how the menus worked and what choices should show up on them.

Due to the fact that NESSUS/EXPERT was undergoing a major change in design, that CLIPS is public domain, portable, and readily accessible from NASA, and that CLIPS could fairly easily and efficiently handle the integration issues, it was decided in March 1987 to reimplement NESSUS/EXPERT in CLIPS and FORTRAN. Though this design philosophy has required an extensive amount of FORTRAN coding, thus slowing development considerably, it has created a highly modular, efficient, and robust user interface to the NESSUS code.

4.1.2 The CLIPS Language

CLIPS is a production rule-based, forward chaining, expert system building tool written in C by a group of individuals at Johnson Space Center [1]. It was developed to meet the needs of systems like NESSUS/EXPERT where speed and integration issues are key to the success of the system. It is the only tool we are aware of that can so completely integrate with other programming environments, including FORTRAN - the programming environment of interest in this effort.

In many ways, CLIPS resembles the expert system building tool used previously in this effort, OPS5. Both use production rules (IF-THEN statements) as their primary means of knowledge representation. Both are forward chaining. That is, they start by gathering data and then make inferences based on this data rather than starting with an inference and trying to find data about the problem that will support that inference. They both use the Rete algorithm for efficient encoding and searching of the production rules in the knowledge base.

In other ways CLIPS differs from OPS5, both in power and functionality. CLIPS provides a much less powerful way of representing data about the domain. It works simply on pattern matching sequences while OPS5

has an actual, though limited, frame representation capability. CLIPS performs conflict resolution using programmer-defined salience factors while OPS5 provides a very nice, implicit method for doing this. On the other hand, CLIPS allows a means of completely integrating the FORTRAN code with the CLIPS rules. In the end, this functional capability out-weighed the disadvantages with respect to power.

4.1.3 The CLIPS/FORTRAN Interface

As stated earlier, the division of work between CLIPS and FORTRAN resulted in separating the higher-level decisions and checks between sets of keywords from the lower-level checks and verifications of parameters and data within a single keyword. CLIPS rules were to be used on the former while FORTRAN routines were to be used to implement the latter. To properly handle each keyword, NESSUS/EXPERT requires a set of FORTRAN routines, C-interface routines, and CLIPS rules.

The integration of CLIPS and FORTRAN can be implemented with either CLIPS or FORTRAN as the "main" program. Development of the system so far has been done with CLIPS as the main program. This arrangement allows for CLIPS to be run in interactive mode, thus providing easier access to CLIPS debugging tools. The main program can easily be changed to FORTRAN if it becomes desirable to do so. Control and communication between CLIPS and FORTRAN is implemented via direct calls to FORTRAN routines or calls to C interface routines which, in turn, invoke the desired FORTRAN routine. The latter is necessary only if parameters are to be passed from CLIPS to FORTRAN. The process of passing parameters to FORTRAN from CLIPS requires the following steps:

1. A C interface routine must be written for each FORTRAN routine that is called with parameters from CLIPS. These C interface routines are simple, the length varying according to the number of parameters being passed. They convert the parameters passed from CLIPS into the C format and then invoke the desired FORTRAN routine.

2. A line of code must be added to a CLIPS routine called USRFUNCS for each C and FORTRAN function called. This line is simply a call to a function called DEFINEFUNCTION with the function name as one of its parameters.

3. To receive the parameters passed from CLIPS into the FORTRAN routine, the parameters must be converted to FORTRAN data types via a call to a CLIPS function called LOADC.

To create objects in the CLIPS world from within a FORTRAN routine, (i.e., passing parameters from FORTRAN back to CLIPS) the data must first be converted to a CLIPS data type and then given to CLIPS. This is accomplished via calls to two CLIPS functions, STOREC and ASSERT.

4.1.3.1 The FORTRAN Side of the Interface

FORTRAN is used to read-in data files or information provided by the user interactively from the keyboard. Based on the keyword that the data is associated with, the FORTRAN routines check for the appropriate number and type of data in each position on each line. Much of this knowledge was acquired from the MHOST Users' Manual [2]. Approximately seven FORTRAN routines must be written for each keyword.

For example, suppose that the user wishes to input data associated with the keyword *ELEMENTS. A top-level FORTRAN routine is used to initiate getting the data, either from a file or directly from the user. Based on where the data is coming from, one of three other routines is then used to actually read-in the data and check it for consistency with respect to the requirements of the keyword in question. Little checking is required for system file input because it is assumed to be correct, having been generated by NESSUS/EXPERT at some previous point. However, user file input or manual entry would require certain verifications. In the case of *ELEMENTS, checks should be made to ensure that the first parameter is a legal element type, and that the subsequent lines of data start with an integer element number followed by the correct number of node numbers for that element type. Formatting restrictions, such as the maximum number of nodes that can occur on a single line (15 in the case of *ELEMENTS) is not checked for here. Rather, the FORTRAN routine that uses this data to create the data deck contains such knowledge.

Salient features of the data are then asserted by FORTRAN into the CLIPS environment via an assert routine. Other routines are needed to get information back from CLIPS, to invoke a help file related to the use of the keyword, and to write the data to a temporary system file and to a NESSUS-readable data deck.

Thus, FORTRAN is used to do all of the complex type checking on all data entered into the system related to a single keyword. CLIPS is not capable of doing certain kinds of type checking, such as integer vs. real, and is much slower at reading large amounts of data into memory.

Though coding such routines in FORTRAN requires more time and effort, once coded the resulting routines are more efficient and effective in this situation.

4.1.3.2 The CLIPS Side of the Interface

Though FORTRAN reads all of the data into memory, none of this information is available to CLIPS without an explicit assertion by the FORTRAN code into the CLIPS environment and a set of CLIPS rules to accept the assertions. Thus, for each keyword there is a set of CLIPS rules that takes the data in from FORTRAN and enters the values into CLIPS data structures. This helps CLIPS keep track of what NESSUS/EXPERT does and does not know about the current problem so far. It also provides the system with the needed information to continue guiding the session (discussed in Section 4.1.4).

Thus, for example, when data about the element types through *ELEMENTS are read-in by FORTRAN, FORTRAN asserts into CLIPS only the total number of nodes for each element type. CLIPS then takes this data and stores it for use during consistency checking between keywords. Other information about the elements may need to be brought into CLIPS at a later time to support certain consistency checking. This will depend on the type of consistency checking that is required and will have to be determined on a case-by-case basis. The goal is to minimize the amount of data that must be passed into CLIPS since if most of the data ends up getting passed, then all of it might as well be read-in, thus slowing the system down.

4.1.4 NESSUS/FEM Interface

The NESSUS/FEM module is a complex finite element code geared toward solving problems with probabilistic data uncertainties. The code uses a newly developed, mixed type formulation, resulting in a new, different computational technology. In order to make this new technology accessible to the users unfamiliar with the code and its theoretical foundations, NESSUS/FEM must be interfaced with an additional code. The role of this new code will be to simplify the use of NESSUS/FEM and to accumulate knowledge on the appropriate usage of the code for various types of problems.

The NESSUS/EXPERT module will serve as an interface to NESSUS/FEM for deterministic analysis. In the complete probabilistic analysis conducted with the aid of NESSUS/FPI, a new module (PFEM) will be used. Its role will be to carry out the algorithms of the probabilistic finite element method and to assure proper information exchange between NESSUS/FEM and NESSUS/FPI. The

PFEM module will be discussed in Section 4.1.6. The new design concept of NESSUS/EXPERT is described in the following subsections.

4.1.4.1 The New Design Concept

The new design concept for the NESSUS/EXPERT system centers around the role of problem database and uses a structured interrogative-interactive mode of operation. The problem database stores all the information about structural problems to be solved, finite element model to be used, random variables to be accounted for, as well as the logistical information about the status of the problem solution process, i.e., if the basic finite element model has been defined, or if any NESSUS/FEM analyses have been run, etc. The information saved in a form of various status indicators, switches and options in the problem database lets NESSUS/EXPERT guide the system user through the solution process by presenting him/with menu selections suitable for a given stage of solution process. For example, probabilistic descript of a problem is not necessary until the determinsitc part of a problem is completed, consequently, the user is not asked to provide probabilistic problem description until it is really needed.

The advantage of this approach lies in the systematic, orderly way the problem is solved. This leads to simplifications in the way the user has to interact with the system (he/she always faces menus that are relevant to the stage of solution at hand, not those that have already been used or those that are not important yet. The new approach markedly simplifies the process of utilization of accumulated knowledge. The information about suitable problem dependent option and parameter selections (determined by accumulated experience) can be conveyed to the user at the most appropriate time, and it can be triggered only as necessary, without overloading the user with excess information.

Also, this step-like approach simplifies internal operations of the system, like model consistency checking, input of user data or preparation of NESSUS/FEM input decks. In this new structured interrogative-interactive approach, NESSUS/EXPERT is always in better control of operations, it does not need to be directed as to what to do next or what data to expect, but it governs the solution process, with the attendant decrease of code complexity and the decrease of need for all encompassing consistency data, parameter and option checking of totally unstructured interaction operation, relying only on user input for control of the solution.

4.1.4.2 An Example Interaction

The prototype version of NESSUS/EXPERT does not have any finite element generation facility. It is assumed that the basic model is normally generated using one of many available general purpose finite element preprocessors (PATRAN, MENTAT, GIFTS, etc.) and the gorups of data such as nodal coordinates, element connectivities, boundary conditions, etc., are stored in separate ASCII files. The proess of building NESSUS/FEM input deck using NESSUS/EXPERT then takes on a form of the following dialog betweent he suer and the program.

The first choice presented to the user by NESSUS/EXPERT is that of starting anew job or resuming one of the existing ones, whose names are listed by the system. If a new job is selected, the user is prompted for a job name and then for the input of the basic model. The basic model definition can be input by providing names of files containing descriptions of nodal coordinates, element connectivities, etc., or by specifying those quantities explicitly. This part of the process is performed in interrogative mode, the system asking specific question and the user providing explicit information (e.g., file name with coordinates or a string of nodal coordinates, etc.). The structural analysis type to be performed is input as part of the basic model description.

As soon as the basic model is defined, the user may input other elements of the problem description, such as material data, loadings, additional elements of the model depending on analysis type, solution control parameters, etc. The mode of the input will be identical as for basic model data. For every category of the input data, the user will be interrogated only for information relevant for the problem at hand. Also, certain guidelines regarding the parameter selection will be presented to the user. The help information will be available onmost menu entries. The "exit/return to main menu" capability will exist in all themenus of the system, allowing the user for an orderly completion of the interactive NESSUS/EXPERT session. The status of the computational model preparation is recorded in the problem database, giving the user a possibility to resume operation from the same stage of the process, at which it was stopped earlier.

Once the full computational model is defined, the NESSUS/FEM input deck is submitted for execution (in batch mode) and the session is completed. The results of analysis are accessible to NESSUS/EXPERT

through PDB file. Upon inspection of the deterministic model results, the user can introduce modifications to the model (for example, to improve accuracy, correct errors, etc.) or he may proceed to define the probabilistic part of the model, and resubmit the modified deck. The process can be repeated until the user is satisfied with the results of deterministic and perturbation analyses, whereupon a full probabilistic analysis using PFEM module may be initiated.

4.1.5. Geometry Perturbation Module

The geometry perturbation module has been developed for generation of perturbations of node coordinates for a typical turbine blade finite element model. The module is oriented for processing turbine blade models built with NESSUS 8-noded solid elements.

4.1.5.1 Perturbation Degrees of Freedom

The perturbation degrees of freedom have been identified based on the vast practical experience of Rocketdyne in the area of SSME turbine blade manufacturing. The identified practically important degrees of freedom are: volume changes (Figure 4.1) translations, and rotations of parts of a blade. All the above perturbation degrees of freedom have been implemented in the module.

The operation of volume change is performed in the global coordinate system (the coordinate system of the finite element model). Translations and rotations can be performed either in the global coordinate system or in any Cartesian local coordinate system specified by the user.

Changes in nodal coordinates, resulting from operations performed upon a model, are accumulated until the user decides to cancel them. This, combined with the capability of storing coordinates of a perturbed model at any time of processing, gives the user maximum flexibility in creating different perturbed versions of an original model.

4.1.5.2 Numerical Implementation

Perturbations of a finite element model are generated in three major stages:

1. The input of coordinates and connectivities of a model from NESSUS deck and the input of model subregion definition from the user. The data from the NESSUS input deck are currently read in a fixed format (upon integration with NESSUS/EXPERT the data will be retrieved from the problem database). The model subregion definition to be provided by the user consists of the number of regions in the model, their names, and the first and last element numbers for every region.

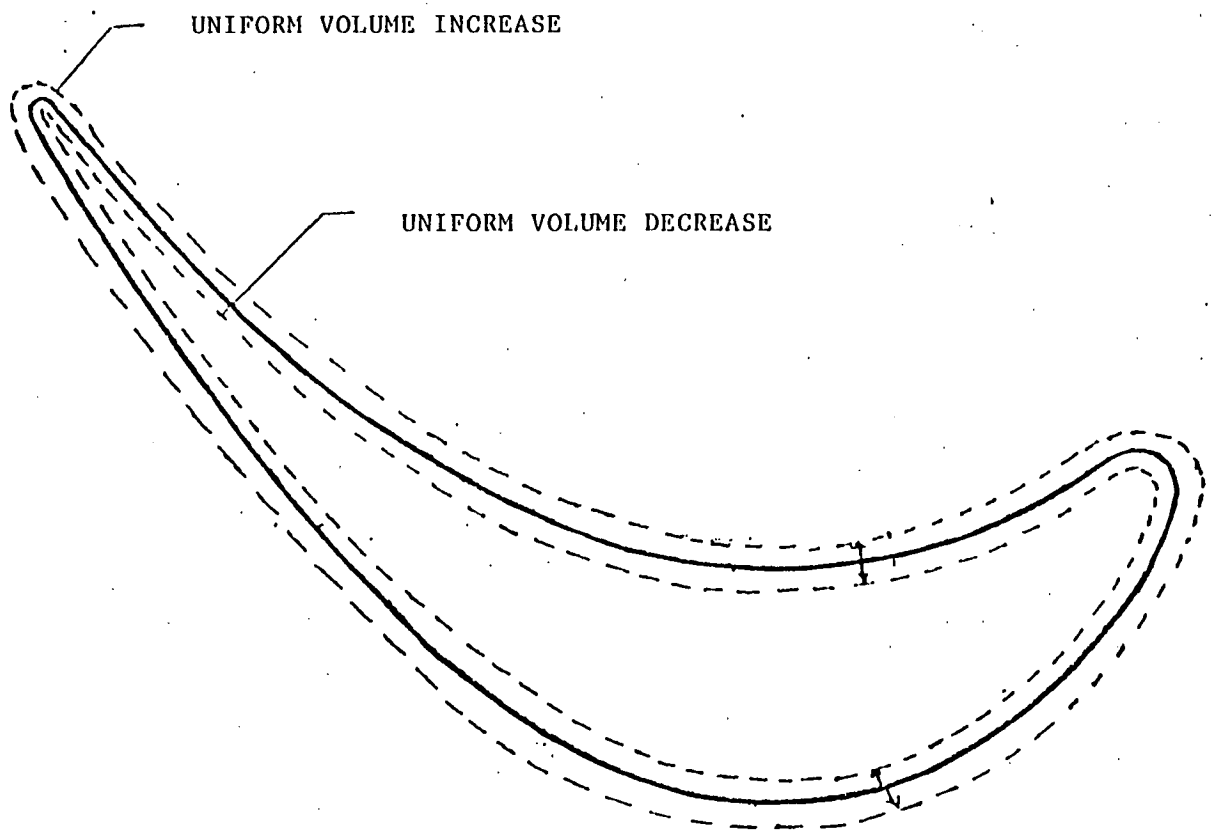


Figure 4.1 Volume Change Degree of Freedom

2. The preparation of the auxiliary geometrical database with such information as node numbers of every region of the model, numbers of surface nodes and walls, and numbers of surface walls attached to every surface node. The operation starts with selecting numbers of nodes associated with every region. Next, the surface walls of elements are identified for the entire model. It is done by checking if a wall belongs to more than one element. If the number of elements containing the wall is equal to one, it means that the wall lies on surface.

   Later, all surface walls are sorted by the numbers of regions to which they belong. At last, the number of surface walls attached to every surface node is calculated.

3. The user data input and the execution of requested operations (volume change, translation, rotation, erase changes, save changes).

Actual changes in nodal coordinates are calculated at this stage. Despite the significant amount of computations required for some of the perturbations (volume change), the response time of the module is still in a reasonable range of up to few seconds, even for the models of large scale (1500 elements, 2500 nodes). This good computational efficiency has been achieved by a careful design of the auxiliary database and the use of such entities as element walls and edges in the surface identification and normal calculation algorithms.

The concept of dynamic storage dimensioning is used in the entire code, making it easy for the analyst to change maximum dimensions allowed inside the code, (it requires changing of appropriate parameters in the main module of the code). The entire code has been written in FORTRAN 77.

4.1.5.3 Mode of Operation

The code is designed to be run interactively. All the necessary information about required input is given to the user through prompts.

The volume change operation requires the following input from the user:

1. region number to which the operation is applied,

2. amount of volume change, measured by the length of a vector normal to the blade surface (+ volume increase, - volume decrease),

3. coordinates of two points defining an auxiliary axis (Figure 4.2),used for defining the surfaces subjected to coordinate changes.

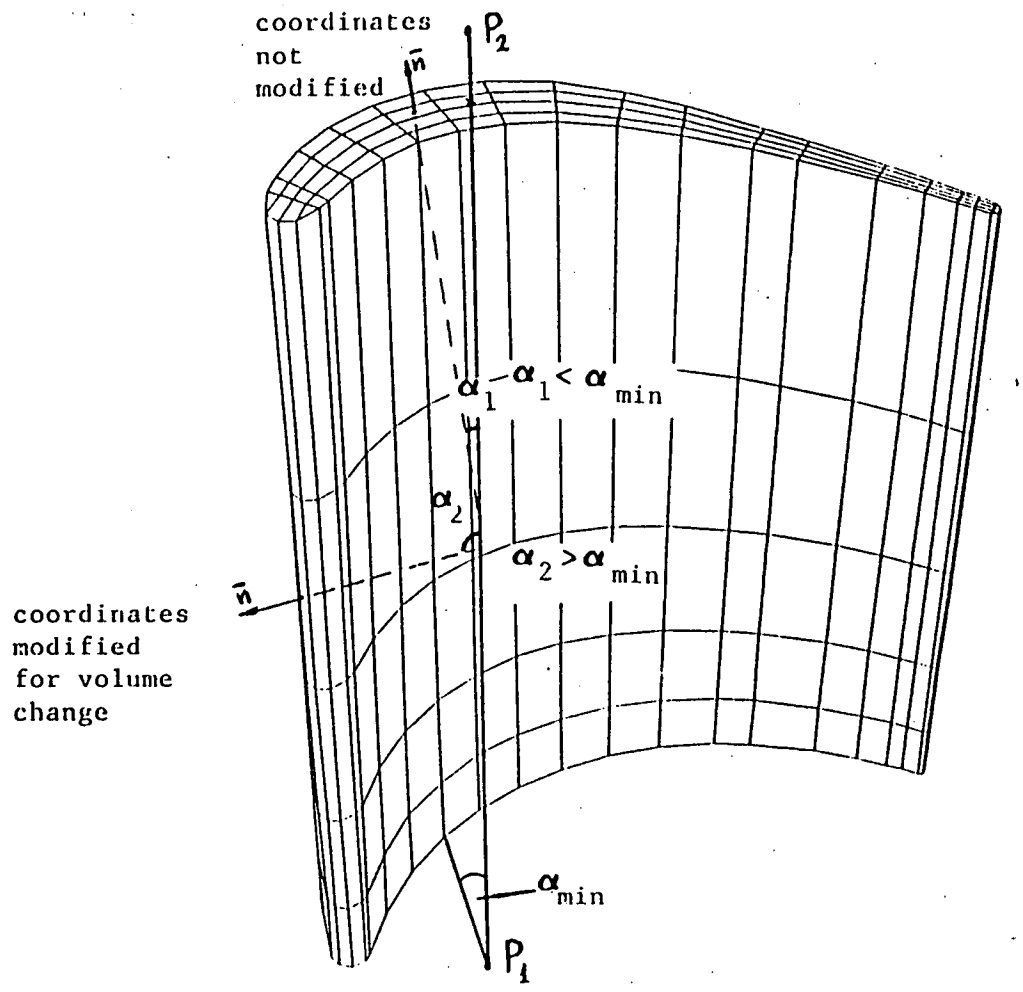


Figure 4.2 Auxiliary Axis P_1-P_2 and Minimum Angle Definition for Volume Change Operation

4. the value of the minimum angle allowed between the auxiliary axis and any of the element normals, (if the angle between an element normal and the the axis is smaller then the minimum angle, then the nodes of the element are not allowed to move in the normal direction).

The operation of translation requires the following input from the user:

1. region number to which the operation is applied,

2. coordinate system in which the operation is performed (if a local coordinate system is selected then the coordinates of three points defining the system are to be input),

3. the values of translations in X, Y, and Z directions of the selected coordinate system.

The operation of rotation requires the following input from the user:

1. region number to which the operation is applied,

2. coordinate system in which the operation is performed (if a local coordinate system is selected then the coordinates of three points defining the system are to be input,

3. the axis number of the coordinate system, about which the rotation is to be performed,

4. the value of the rotation angle (in degrees)

4.1.6 NESSUS/PFEM Module

The NESSUS/PFEM module has been designed as a batch mode program for the Probabilistic Finite Element Method (PFEM). The principal function of the program is to perform complete probabilistic analysis of the problem using both the NESSUS/FEM and the NESSUS/FPI modules. The function is accomplished by repeated alternate executions of both modules, accounting for various types of probabilistic analysis and/or possible numerical problems with perturbation analysis. The batch mode of operation has been selected because of long run times of NESSUS/EXE module for computational models of practical size. The input data for the NESSUS/PFEM module is prepared during an interactive session with NESSUS/EXPERT.

Detailed descriptions of the NESSUS/PFEM module follow.

4.1.6.1 Types of Probabilistic Analysis

There are two basic types of probabilistic analysis available in the PFEM module. The first one, named global, evaluates the global

response of a performance function (stress at a point, at various levels of probability). In other words, it gives an overall variation of the probability versus the performance function levels, over the range of practically attainable performanc function values.

The second type of analysis is named local, since it is concerned with more "local" behavior of the performance function. There are two kinds of local probabilistic analysis: where the performance function level is calculated for a given probaility level, and where the probability level is calculated for a specified value of performance function.

The global analysis is performed in two basic steps. The first step consists of global mean-value-first-order (MVFO) analysis using the NESSUS/FPI code and the FEM perturbation data. In this step, design point coordinates are calculated at 9 ÷13 probability levels covering the range of practical interest ($0.00001 < p < 0.99999$). In the second step, the NESSUS/FEM code is used to calculate the performance function values for design point coordinates calculated in the first step. It is assumed that the probability levels corresponding to design points are accurate and the performance function values calculated in the second step constitute a final solution (Figure 4.3). No iteration perturbations are performed at the final probability performance funciton levels. Practical experience showed that the improvement of the solution is small in such situations so that the more accurate, but also more expensive, iterative approach is used only for local analyses.

The local analysis for specified probability level utilizes the newly developed FPI code capability of calculation of performance function and design point values for a given probability value. The algorithm for this type of analysis starts with MFVO FPI run to determine the design point coordinates and performance function value for specified probability level. The subsequent recomputations of performance function and perturbation analysis around the design point in NESSUS/FEM is used to iterate for accurate value of performance function (Figure 4.4).

The local analysis for specified performance function value is more complex. The first step of the algorithm evaluates a crude approximation to the probability level and design point coordinates corresponding to specified performance function value, using the MVFO FPI run. A more refined approximation to the probability level and design point

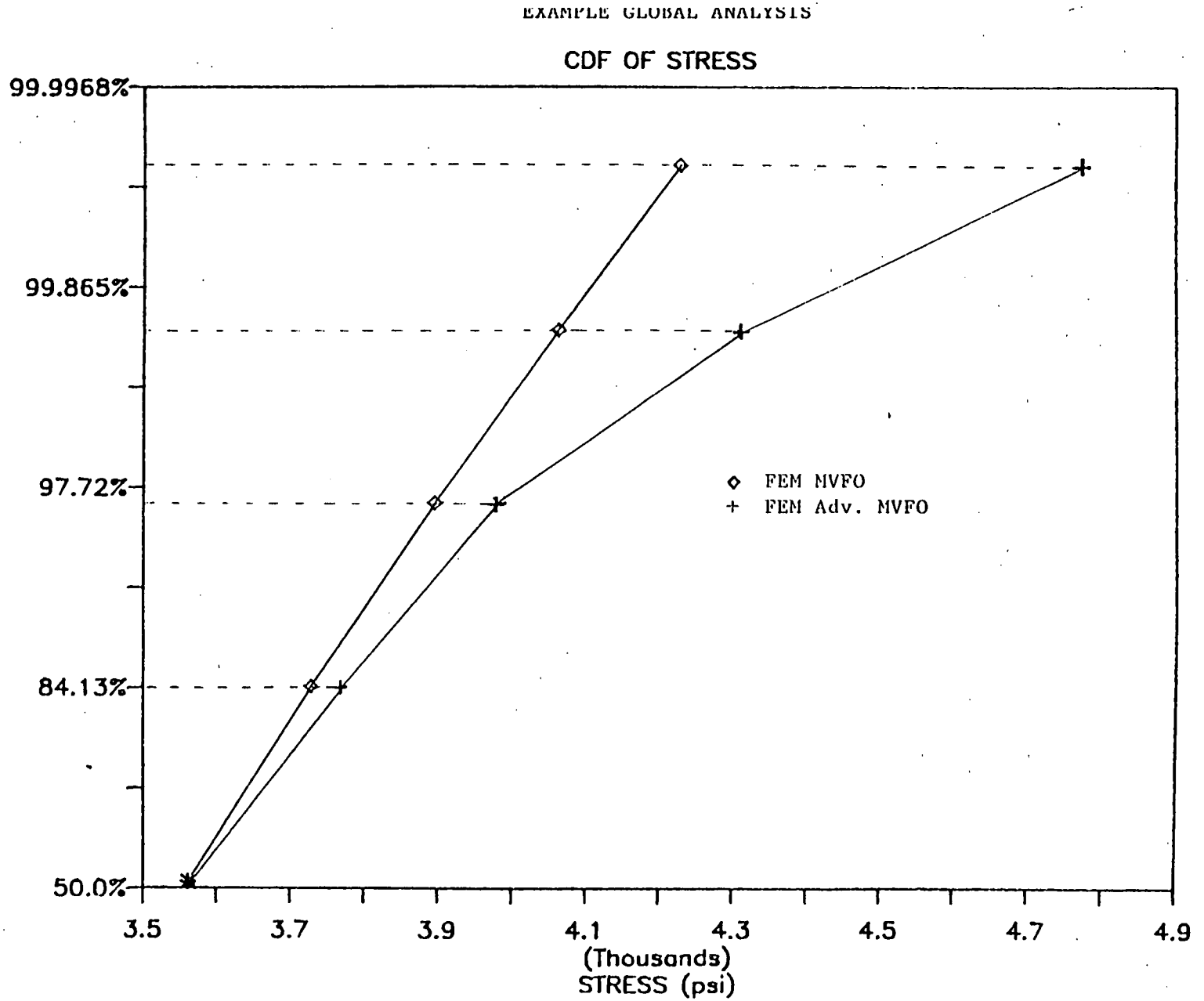


Figure 4.3 Global Probabilistic Analysis Results

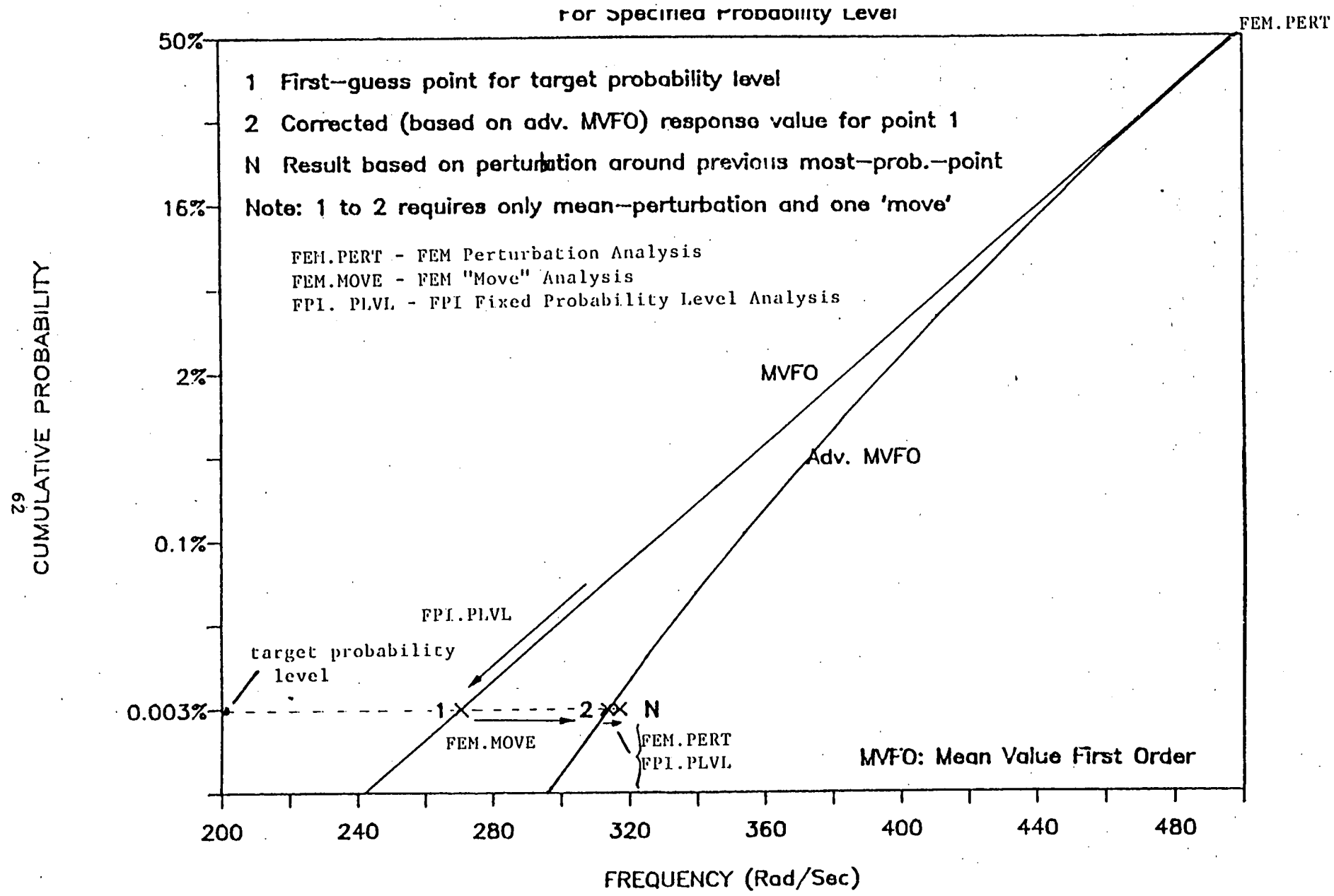


Figure 4.4 Local Analysis with Specified Probability Level

coordinates is obtained by recalculating the performance function (NESSUS/FEM) at a previous level of probability and using quadratic interpolation. Once a good approximation to the design point coordinates is found, iterations using perturbation (FEM) data about that point are used to locate the final solution (Figure 4.5).

In the prototype version of the PFEM module only the global analysis is currently implemented.

4.1.6.2 Transfer of Information Between FEM and FPI Modules

There is a significant transfer of information between the FEM and FPI modules of NESSUS in the process of probailistic analysis. The NESSUS/FEM module provides the values of performance function: stress, strain, displacement frequency, etc., for specified fixed values of random variables (geometry, material, loading parameters). Also, the FEM module provides information about success or lack thereof in the solution process, which information is later used in appropriate corrective actions. The NESSUS/FPI module provides the values of design point coordinates (values of random variables) and their corresponding probability levels, along with their estimated performance function values.

The above information is transferred between the modules in form of files. The output from NESSUS/FEM is stored in the perturbation database (file PDB). The NESSUS/FPI output is passed to the PFEM module executive through a coded file with extension FPO. All of the files passed within PFEM have a common first part of filename and are treated as a part of Problem Database.

4.1.6.3 Interaction with Database

The PFEM module is designed in such a fashion that it receives very little data directly from the NESSUS/EXPERT code. The data passed to PFEM is limited to a few control parameters, defining type of anlaysis, identifying random variables and performance functions, etc. The bulk of input to PFEM is contained in the Problem Database. The information stored here is used to assemble both NESSUS/FEM and NESSUS/FPI input decks. On the other hand, all the intermediate problem data used by PFEM, as well as the final results, are also stored in the Problem Database. This arrangement makes it possible for NESSUS/EXPERT to access the status and the results of the analysis and present them to the user.

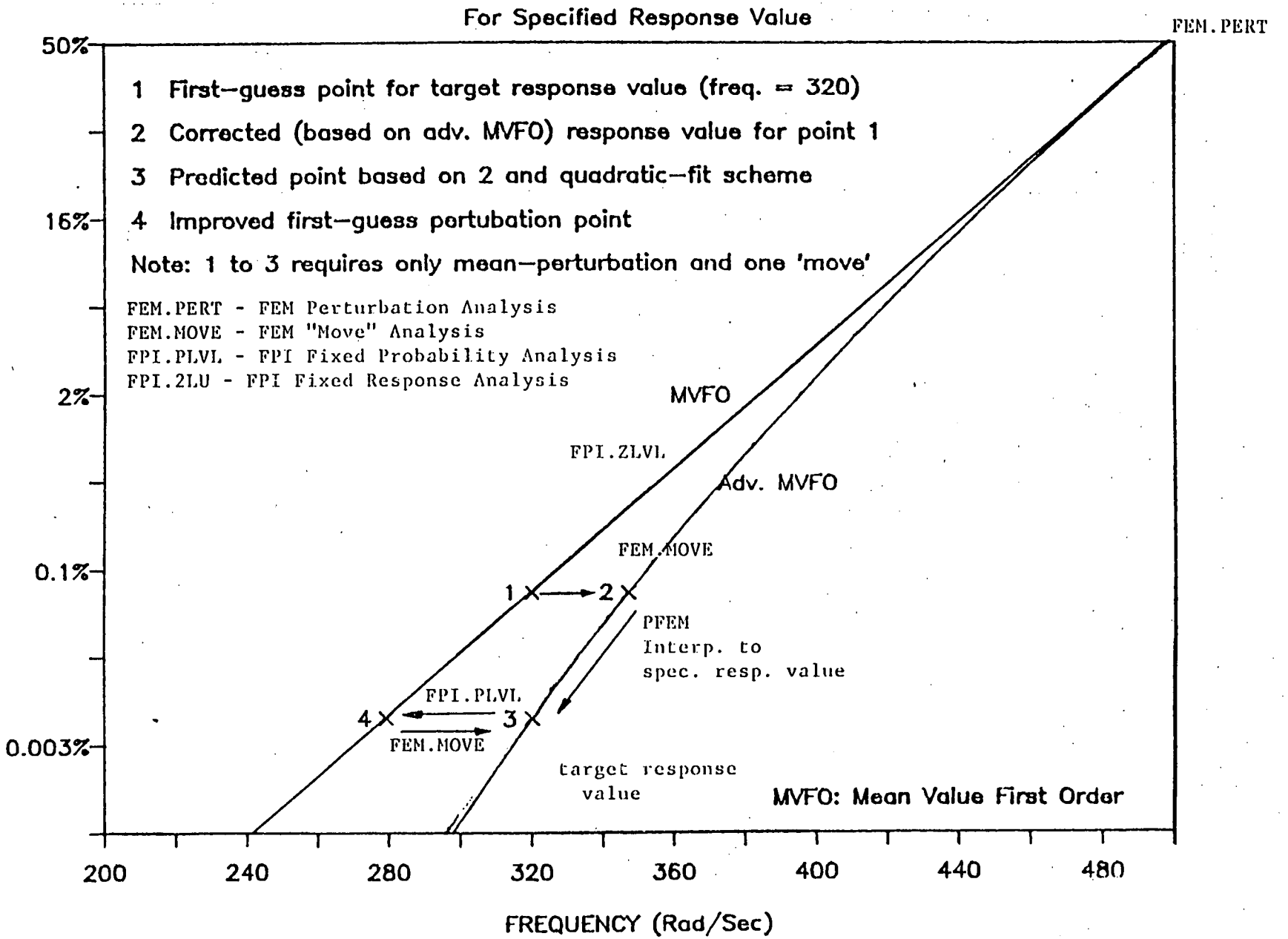


Figure 4.5 Local Analysis with Specified Response Value Level

The interaction of the PFEM module with Problem Database is very extensive. The complete description of the finite element model along with random variable definitions are used in PFEM to build various perturbed variable models for NESSUS/FEM. Results of the FPI analyses, in terms of design point coordinates (and corresponding probability levels) are stored in Problem Database for later reuse in more accurate estimates of performance function values. Generally, all the information obtained in the course of analysis that is important from the point of view of further analysis (essential intermediate results and experience gathering (computational process efficiency measures), is saved in the Problem Database for later access.

4.1.7 NESSUS/FPI Interface

Nothing has been done on this portion of NESSUS/EXPERT to date.

4.2 Current Efforts on NESSUS/EXPERT

At the end of FY87, the initial NESSUS/FEM interface in NESSUS/EXPERT was nearing completion. Another month of effort will result in a prototype system ready for evaluation. The system will know about approximately 60 keywords used to run NESSUS/FEM and will have a small amount of knowledge acquired through the experience of running NESSUS/FEM. The experiential knowledge will grow for the duration of the project. This will involve maintaining records or experience gained from using NESSUS/FEM and embodying as much of the experience as possible into CLIPS rules.

After completion of this initial NESSUS/EXPERT for deterministic analysis, efforts will turn to the development of the interface for the probabilistic portion. A basic design concept should be agreed upon before implementation begins. The plan is to have a completed version of the probabilistic portion of NESSUS/EXPERT during the third quarter of FY88. The system will then be distributed for evaluation as will be done with the deterministic portion.

References

[1] Chris Culbert, "CLIPS Reference Manual - Version 4.0," Mission Support Directorate, MPAD, NASA-JSC, March 1987.

[2] S. Nakazawa, J.B. Dias, J.C. Nagtegaal, and T.B. Wertheimer, "MHOST Users' Manual - Version 3.3," April 1986.

5.0 NESSUS BOUNDARY ELEMENT CODE DEVELOPMENT

5.1 Summary

This section describes the development of a boundary element method (BEM) formulation for probabilistic stress analysis. The BEM contrasts with domain methods such as finite element method for linear problems by the fact that the governing integral equations (called the Boundary Integral Equation or BIE) are expressed over the boundary of the body [1-3]. The essential feature of the boundary element method is the availability of singular (fundamental) solutions of the governing equilibrium equation. In principle, the probabilistic boundary element formulation requires the solution of stochastic equilibrium equations, which does not appear to be available for a general case. The approach used herein is to extract the probabilistic results from the deterministic solution.

For problems with body forces such as thermoelastic and transient loading problems, a direct transformation of the equilibrium equations to integral equations over the surface of the body is generally not possible. The inhomogeneous part of the governing equations will appear as a particular integral over the domain of the body. Further, to obtain the probabilistic solutions, the deterministic problem is solved repeatedly for each perturbation of random variable. Therefore, efficient deterministic BEM formulations are sought for the current analysis. One of the major features of the current analysis is that the domain integrals are transformed through certain approximations such that the resulting BIE is expressed over the boundary of the domain only.

Further, the probabilistic results are obtained from the deterministic solutions through perturbation of random variables. Efficient algorithms for the determination of perturbed solution variables are also discussed.

5.2 Probablistic BEM Formulation

The governing equilibrium equation can be transformed through the use of the fundamental solution to integral equations over the surface for homogeneous, elastic, isotropic bodies in the absence of body forces. For nonlinear and general body force problems, such a surface transformation, in general, is not possible. The resulting integral equation will consist of a particular integral over the domain of nonlinearity or inhomogeneity.

5.2.1 Governing Equations

Consider the equilibrium of an element of a body under thermoelastic transient loading conditions. Using D'Alembert's principle, the equilibrium equation can be expressed as

$$\sigma_{ij,j} + b_i - \rho \ddot{u}_i = 0 \tag{5.1}$$

where, σ_{ij}, is the stress tensor, b_i, is the body forces vector, u_i, is the displacement vector, ρ, is the density, and superior dot indicates derivative with respect to time. The stresses are related to strains through the thermoelastic constitutive relationship [4] as

$$\sigma_{ij} = D_{ijkl}\left(\epsilon_{kl} - \epsilon^{\theta}_{kl}\right) \tag{5.2}$$

where D_{ijkl} is the (temperature dependent) elastic constitutive matrix given in terms of shear modulus, μ, and Poisson's ratio, ν, as

$$D_{ijkl} = \frac{2\nu\mu}{1-2\nu}\delta_{ij}\delta_{kl} + \mu\left(\delta_{ik}\delta_{jl} + \delta_{jk}\delta_{kl}\right) \tag{5.3}$$

The total strain, ϵ_{ij}, and thermal strain, ϵ^{θ}_{ij}, are given by

$$\epsilon_{ij} = \frac{1}{2}\left(u_{i,j} + u_{j,i}\right), \tag{5.4}$$

$$\epsilon^{\theta}_{ij} = \alpha\delta_{ij}\theta \tag{5.5}$$

where, α, is the temperature dependent coefficient of thermal expansion, and, θ, is the change in temperature from unstressed state.

Let us define u^{θ}_i such that

$$\epsilon^{\theta}_{ij} = \frac{1}{2}\left(u^{\theta}_{i,j} + u^{\theta}_{j,i}\right) \tag{5.6}$$

and

$$\hat{u}_i = u_i - u^{\theta}_i \tag{5.7}$$

The stress-strain relationship can then be written as

$$\sigma_{ij} = D_{ijkl}\hat{u}_{k,l} \tag{5.8}$$

Consider an auxiliary field with homogeneous material properties $\bar{\mu}$, $\bar{\nu}$ and corresponding constitutive matrix $\bar{D}_{ijkl}$. The fundamental solution, u^p, due to unit point force, e_i, for the field is well known and is evaluated from

$$\bar{D}_{ijkl} u^p_{k,lj} = \delta e_i \tag{5.9}$$

where δ is the Dirac delta. Let us define an 'initial stress', σ^i_{ij} , as

$$\sigma^i_{ij} = \sigma^h_{ij} - \sigma_{ij} \tag{5.10}$$

where,

$$\sigma^h_{ij} = \bar{D}_{ijkl} \hat{u}_{kl} \tag{5.11}$$

Then equation (5.8) can be expressed as

$$\sigma_{ij} = \bar{D}_{ijkl} \tilde{u}_{k,l} \tag{5.12}$$

$$\tilde{u}_i = \hat{u}_i - u^i_i \tag{5.13}$$

and u^i is given by

$$\sigma^i_{ij} = \bar{D}_{ijkl} u^i_{k,l} \tag{5.14}$$

Using the relationship (5.12), the governing equilibrium equation can be expressed as

$$\bar{D}_{ijkl} \tilde{u}_{k,lj} = -b_i + \rho \ddot{u}_i \tag{5.15}$$

Let us define a stress field σ^d_{ij} corresponding to displacement field u^d that satisfies the inertial part of stresses. i.e.

$$\sigma^d_{ij,j} = \rho \ddot{u}_i \tag{5.16}$$

and

$$\sigma^d_{ij} = \bar{D}_{ijkl} u^d_{k,l} \tag{5.17}$$

The governing equilibrium equation can then be expressed as

$$\bar{\sigma}_{ij,j} = -b_i \tag{5.18}$$

where,

$$\bar{\sigma}_{ij} = \bar{D}_{ijkl}\bar{u}_{k,l} \tag{5.19}$$

$$\bar{u}_i = \bar{u}_i - u_i^d \tag{5.20}$$

5.2.2 BIE Formulation

The classical direct boundary integral equation is obtained by applying divergence theorem to the product of the equilibrium equation (5.18) and u_i^p within the domain Ω as

$$\int_\Omega \bar{\sigma}_{ij,j}.u_i^p\, d\Omega = -\int_\Omega b_i.u_i^p\, d\Omega \tag{5.21}$$

which reduces to

$$\bar{u}_j + \int_\Gamma T_{ij}\bar{u}_i\, d\Gamma - \int_\Gamma U_{ij}\bar{t}_i\, d\Gamma = \int_\Omega b_i U_{ij}\, d\Omega \tag{5.22}$$

where Γ is the surface of the domain, U_{ij} and T_{ij} are displacement and traction point force solutions, $\bar{t}_i$ is given by

$$\bar{t}_i = t_i - t_i^d \tag{5.23}$$

$$t_i^d = \sigma_{ij}^d n_j \tag{5.24}$$

and n_j is the normal vector at the surface.

The above integral equation (5.22) still contains the domain integral of the body force vector. Other than thermal, inertial and inhomogeneity body forces, which have been taken into account already in the analysis, the only other body force considered in the present analysis is the centrifugal loading. The centrifugal body force again can be treated by the procedure described earlier. However, the domain integral due to body forces with potential such as the centrifugal load can be converted to surface integrals as described in the next section.

5.2.3 Body Force with Potential

The domain integral due to body forces is given by

$$B_j = \int_\Omega U_{ij} b_i \, d\Omega \tag{5.25}$$

The body force vector due to the rotation of the body about an axis through the origin of coordinates with an angular velocity ω_i can be expressed as

$$b_i = R_{im} x_m \tag{5.26}$$

where x_m is any point within the domain,

$$R_{im} = -\rho e_{ijk} \omega_j e_{klm} \omega_l, \tag{5.27}$$

and e_{ijk} is the permutation tensor.

Further, the fundamental solution can be expressed in terms of Galerkin vector, G_{ij}, as

$$U_{ij} = G_{ij,kk} - \frac{1}{2(1-\nu)} G_{ik,kj} \tag{5.28}$$

$$G_{ij} = \frac{1}{8\pi\mu} \delta_{ij} r \tag{5.29}$$

where r is the distance between source and field points.

By substituting equations (5.26) and (5.28) into (5.29) and integrating by parts, we have

$$B_i = \int_\Gamma P_i \, d\Gamma \tag{5.30}$$

where P_i is given in Appendix D (equation D-1). The transformation procedure described in this section follows previous works given in [5-7].

5.2.4 Numerical Implementation

The boundary integral equation corresponding to (5.22) at the surface can be derived by treating the resulting singular integrals appropriately [1-3]. To solve these equations, the body is divided into arbitrary boundary elements over which the geometry as well as field variables are approximated by interpolation functions. Upon the evaluation of the discretized integrals, the equations can be assembled to form a system of equations, expressed in matrix form as

$$\underline{U}\,\underline{\dot{t}} = \underline{T}\,\underline{\dot{u}} + \underline{b} \tag{5.31}$$

or

$$\underline{U}\,\underline{t} - \underline{T}\,\underline{u} = \underline{U}\,\underline{t}^{\circ} - \underline{T}\,\underline{u}^{\circ} + \underline{b} \tag{5.32}$$

where

$$u_i^{\circ} = u_i^{\theta} + u_i^{i} + u_i^{d} \tag{5.33}$$

$$t_i^{\circ} = t_i^{d} \tag{5.34}$$

The solution to the above equations requires knowledge of u_i° and t_i°. These terms are evaluated by solving previously defined equations as described in the following section.

5.3 Body Force Interpolation Algorithm

As described in the previous section, the body forces due to thermal and transient loadings are transformed to the surface through particular solutions of the displacement fields of the inhomogeneous equations. The success of the procedure depends on the feasibility of obtaining particular solutions to the governing equations.

5.3.1 Thermal Body Force Analysis

A particular solution to u_i^{θ} can be determined by the solutions of equations (5.5) and (5.6). i.e.,

$$\frac{1}{2}\left(u_{i,j}^{\theta} + u_{j,i}^{\theta}\right) = \alpha \delta_{ij} \theta \tag{5.35}$$

Since the solution requires the knowledge of the temperature field, an assumption is made regarding the temperature distribution. A convenient way is to represent the temperature field by a function of the form

$$\bar{\theta}(P) = \alpha(P)\theta(P) = K\left(P, Q_m\right)\phi^{\theta}\left(Q_m\right) \tag{5.36}$$

where $K(P,Q_m)$ is an assumed function, $\phi^{\theta}(Q_m)$ is an unknown coefficient associated with point Q_m, and summation is implied over subscript m. A solution for the displacement is obtained by satisfying equation (5.35) as

$$u_i^{\theta}(P) = G_i^{\circ}\left(P, Q_m\right)\phi^{\theta}\left(Q_m\right) \tag{5.37}$$

where expressions for K and c_i are given in Appendix D (equations D-2 & D-3).

The accuracy of the procedure depends on how well the temperature field is approximated by equation (5.36). The temperature distribution for the problems considered in the current project will have a high thermal gradient at the surface of the body. The global approximation described above may not adequately represent this variation unless a large number of sampling points are selected near the surface, which makes the procedure inefficient. One way to enhance the procedure is to use a different scheme for the near surface temperature analysis. Let the temperature field be decomposed into two parts,

$$\theta = \theta^{(1)} + \theta^{(2)} \tag{5.38}$$

where $\theta^{(1)}$ is a one-dimensional field varying exponentially normal to the surface of the body as

$$\alpha\theta^{(1)} = \theta^{\circ} e^{-\frac{\xi}{L-\xi}} \tag{5.39}$$

In equation (5.39), ξ is the normal distance (referred to a local coordinate system constructed at the boundary point) and, L is the distance over which this exponential temperature variation is assumed to occur. A displacement field satisfying this conditions can be derived in terms of a displacement potential, ψ, as

$$u_i^{(1)} = \psi_{,i} \tag{5.40}$$

where ψ is given in Appendix D (equation D-4). The overall displacement solution is then obtained as

$$u_i^{\theta} = u_i^{(1)} + u_i^{(2)} \tag{5.41}$$

where $u_i^{(2)}$ is obtained from equation (5.35) by replacing θ by $\theta-\theta^{(1)}$.

5.3.2 Temperature Dependent Material Properties Analysis

The Inhomogeneity arising from temperature dependent material properties may be analyzed by a similar procedure. A displacement solution due to material inhomogeneity can be determined from equation (5.14). Assuming that σ_{ij}^i may be interpolated by a generalized function, the corresponding displacement solution is evaluated as

$$\sigma_{ij}^i(P) = K(P, Q_m)\phi_{ij}^i(Q_m) \tag{5.42}$$

$$u_k^i(P) = G_{ijk}^i(P, Q_m)\phi_{ij}^i(Q_m) \tag{5.43}$$

where G_{ijk}^i is given in Appendix D (equation D-5).

5.3.3 Transient Analysis

Transient problems may also be analyzed by the above procedure. In this report we consider free vibration analysis only and a displacement solution can be determined from equations (5.16) and (5.17); i.e.

$$D_{ijkl}u_{k,lj}^d = -\omega^2 \rho u_i \tag{5.44}$$

where ω is the natural frequency. Representing ρu_i by a generalized function, a displacement solution that satisfies the above equation can be determined.

$$v_i(P) = \rho(P)u_i(P) = K(P, Q_m)\phi_i^d(Q_m) \tag{5.45}$$

$$u_j^d(P) = \omega^2 G_{ij}^d(P, Q_m)\phi_i^d(Q_m) \tag{5.46}$$

Using kinematic and constitutive relationship, the corresponding traction solution can be evaluated as

$$t_j^d(P) = \omega^2 H_{ij}^d(P, Q_m)\phi_i^d(Q_m) \tag{5.47}$$

where G_{ij}^d, and H_{ij}^d are given in Appendix D (equation D-6 and D-7). A similar procedure for problems with constant material density is given in [8,9].

5.3.4 Deterministic Solution Algorithm

The boundary values of displacements and tractions are obtained by solving equation (5.32) satisfying prescribed boundary loading. Substituting the particular solutions for displacements and tractions we have

$$\underline{U}\,\underline{t} - \underline{T}\,\underline{u} = \underline{U}\left[\omega^2 \underline{H}^d \underline{\phi}^d\right] - \underline{T}\left[\left(\underline{G}^\theta \underline{\phi}^\theta + \underline{\psi}\right) + \underline{G}^i \underline{\phi}^i + \omega^2 \underline{G}^d \underline{\phi}^d\right] + \underline{b} \tag{5.48}$$

where the unknown coefficients $\underline{\phi}^\theta$, $\underline{\phi}^i$, and $\underline{\phi}^d$ are related to temperature, initial stress, and displacement fields by equations (5.36), (5.42), and (5.45). The straight forward approach for determining the unknown coefficients is to choose Q_m to coincide with the boundary nodes. Matrices corresponding to these

equations can be made square by collocating at the same number of points as the boundary nodes. The equations can then be inverted to obtain the following relationships:

$$\underline{\phi}^{\theta} = \underline{F}^{\theta}\underline{\theta} \tag{5.49}$$

$$\underline{\phi}^{i} = \underline{F}^{i}\underline{\sigma}^{i} \tag{5.50}$$

$$\underline{\phi}^{d} = \underline{F}^{d}\underline{u} \tag{5.51}$$

where $\underline{F}^{\theta}$, $\underline{F}^{i}$, and $\underline{F}^{d}$ are defined in Appendix D (equations D-8, D-9 and D-10). The system equation is then reduced to

$$\underline{A}\underline{x} - \underline{f} = \omega^2\left(\underline{A}^{d}\underline{x} - \underline{f}^{d}\right) + \underline{T}^{\theta}\underline{\theta} + \underline{T}^{i}\underline{\sigma}^{i} + \underline{b} \tag{5.52}$$

where $\underline{x}$ is the vector of unknown boundary displacement and tractions, $\underline{f}$ is the vector due to applied mechanical loading, $\underline{b}$ is the vector due to centrifugal body forces , and $\underline{A}$, $\underline{A}^{d}$, $\underline{T}^{\theta}$, $\underline{T}^{i}$ are obtained from matrix manipulation.

Mechanical and Centrifugal Loading Solution Algorithm

For mechanical and rotational loading cases, equation (5.52) can be reduced to

$$\underline{A}\underline{x} = \underline{f} + \underline{b} \tag{5.53}$$

and the solution to this equation is straightforward.

Thermal Loading Solution Algorithm

For thermal loading, equation (5.52) can be reduced to

$$\underline{A}\underline{x} = \underline{T}^{\theta}\underline{\theta} + \underline{T}^{i}\underline{\sigma}^{i} \tag{5.54}$$

The initial stress in the above equation can be evaluated from

$$\sigma^{i}_{ij} = \left(\overline{D}_{ijkl} - D_{ijkl}\right)\hat{\epsilon}_{kl} \tag{5.55}$$

where

$$\hat{\epsilon}_{kl} = \epsilon_{kl} - \epsilon^{\theta}_{kl} \tag{5.56}$$

The displacement gradient can be obtained from the derivatives of equation (5.22). Since the evaluation of displacement gradients requires complete

knowledge of boundary displacements and tractions, some form of iterative procedure is necessary for coupling the solutions of equations (5.54) and (5.55).

Free Vibration Solution Algorithm

For free vibration analysis, equation (5.52) is reduced to

$$\left[\underline{A} - \omega^2 \underline{A}^d\right]\underline{x} = \underline{0} \tag{5.57}$$

Using eigenvalue extraction routines, the above equation can be solved.

5.3.5 Perturbation Solution Algorithm

The boundary element formulation and solution procedures described in the previous sections pertain to deterministic systems. The probabilistic structural response is determined by applying FPI to the sensitivities of response variables. The evaluation of the sensitivities requires repeated calculation of response parameters due to the perturbation of random variables. Since the substantial portion of the computational effort is spent for these evaluations, an efficient algorithm is essential for the method to be used as a practical solution tool.

The boundary integral equations derived earlier are for the unperturbed system. The system equation (5.52) can be expressed as a function of random variables vector χ. For quasi-static loading, the perturbed system equation can be expressed as

$$d(\underline{A}\underline{x}) - d\underline{f} = d\underline{b} + d\left(\underline{T}^{\theta}\underline{\theta}\right) + d\left(\underline{T}^{i}\underline{\sigma}^{i}\right) \tag{5.58}$$

Loading Perturbations

The randomness of applied mechanical and centrifugal loading will reduce equation (5.58) to

$$\underline{A}\Delta\underline{x} = \Delta\underline{f} + \Delta\underline{b} \tag{5.59}$$

The perturbation solution of the response variable is then obtained by solving a system of equations with the same matrix as the one in the determined case. In the presence of thermal loading, equation (5.58) is reduced to

$$\underline{A}\Delta\underline{x} = \underline{T}^{0}\Delta\underline{\theta} + \underline{T}^{i}\Delta\underline{\sigma}^{i} \tag{5.60}$$

and equation (5.55) is reduced to

$$\Delta\underline{\sigma}' = -\Delta\underline{D}\,\hat{\underline{\epsilon}} + (\overline{\underline{D}} - \underline{D})\Delta\hat{\underline{\epsilon}} \tag{5.61}$$

where $\Delta\hat{\underline{\epsilon}}$ can be determined from the perturbed displacement gradient equation corresponding to (5.22).

Geometry Perturbations

The system equations corresponding to geometric perturbations can be deduced from equation (5.58) as

$$\underline{A}\,\Delta\underline{x} = \Delta\underline{f} + \Delta\underline{b} + \Delta\underline{A}\,\underline{x} + \Delta\underline{T}^{\theta}\underline{\theta} + \Delta\underline{T}^{i}\underline{\sigma}^{i} + \underline{T}^{i}\Delta\underline{\sigma}^{i} \tag{5.62}$$

Since the same matrix as before is solved, the solution can be evaluated efficiently, provided $\Delta\underline{f}$, $\Delta\underline{b}$, $\Delta\underline{A}$, $\Delta\underline{T}^{\theta}$, and $\Delta\underline{T}^{i}$ can be computed effectively.

Material Properties Perturbations

The change in stress due to changes in the material properties can be conceived as a form of initial stress (σ^{m}_{ij}). We can define such an initial stress as

$$\sigma^{m}_{ij} = \left(\overline{D}_{ijkl} - D^{m}_{ijkl}\right)\hat{\epsilon}_{kl} \tag{5.63}$$

where D^{m}_{ijkl} is the constitutive matrix corresponding to the perturbed material properties and

$$\hat{\epsilon}_{kl} = \epsilon_{kl} - \epsilon^{m}_{kl} \tag{5.64}$$

A system equation can be formed following the procedures described for thermal inhomogeneities as

$$\underline{A}\,\underline{x} = \underline{T}^{m}\underline{\sigma}^{m} \tag{5.65}$$

The perturbed equation for the material properties can be deduced from the above equation as

$$\underline{A}\,\Delta\underline{x} = \underline{T}^{m}\Delta\underline{\sigma}^{m} \tag{5.66}$$

where $\Delta\underline{\sigma}^{m}$ may be evaluated from appropriate derivatives of equations (5.22) and constitutive equation (5.63). Again, as with the temperature dependent material properties solution algorithm, an iterative procedure is necessary. For homogeneous bodies, the perturbation algorithm for the material properties may be simplified such that neither interior displacement derivative solution nor an iterative procedure is necessary.

5.4 Status and Future Plans

The effort in the reporting period mostly concerned with the theoretical development of a boundary element formulation for probabilistic stress analysis. The computer code for general stress analysis including eigenvalue analysis was developed from BEST3D code. A limited number of simple test problems were run using this code. The computer program is yet to be developed for thermal analysis that includes temperature dependent material properties. A perturbation solution algorithm is also not incorporated in the computer code.

The next year effort will mostly cover the completion of the programming of the algorithms discussed so far and to continue the development of BEM formulation. For the isotropic deterministic case, the BEM formulation is mostly complete. An efficient way to evaluate perturbed eigenvalue extraction is yet to be developed. Further, additional investigation is needed for efficient geometric perturbation analysis. Some of the analyses may be simplified considerably for homogeneous bodies. An investigation into using a simplified procedure for some specific cases will be completed during the next year.

Only linear problems have been considered in this report. Once the linear analysis is completed, the computer code will be validated using a number of sample problems. The code will then be included in the NESSUS framework. Further, a data base consistent with NESSUS/FEM will be developed for subsequent statistical analysis. Interface for NESSUS/EXPERT will also be developed.

Even though the BEM formulation developed here is for isotropic materials, the formulation for most part can be used for anisotropic materials. However, a closed form solution for the single crystal anisotropic material used in this project is not available. To use the algorithm developed for the isotropic material to anisotropic case, some form of approximate solutions needs to be developed. The next year effort will also focus on such development. In addition, approximate nonlinear modeling strategies will be investigated.

6.0 NESSUS CODE VALIDATION STUDIES

6.1 Overview of Code Validation Efforts

A plan for validating the NESSUS probabilistic finite element code was included in the PSAM First Year Annual Report (Vol. III, Section 4). The original plan consisted of nine validation problems. During the last year, the number of the problems has increased to fourteen (see Table 6.1) to test other capabilities of the NESSUS code.

Exact solutions, in terms of probability distributions or the probability of exceedance, have been obtained for validation problems numbers 1 to 7, 9 and 10. NESSUS validations were successfully completed for this problem set except for problems 4 and 5. A summary of the validation problems completed in FY '87 and the problems to be completed in FY '88 is listed in Table 6.1. Note that, except for problem 8, problems to be addressed in the next year are those which could not be solved using the NESSUS version 2.0. The recently released NESSUS 2.5 version will be capable of solving problems 4 and 5 (rotating beam and rotating plate).

The results for the completed validation studies are presented in the following sections. More detailed summaries of the validation cases are documented in Appendix A using a "standard format." The standard format was designed to include all the required input data and information. In addition to validating the code, a new user can use these problems to gain confidence that he is using the code correctly.

When closed-form probability solutions are not available, exact solutions were obtained by using Monte Carlo simulation. The "exact" solutions were compared with NESSUS results to validate the code as well as the solution algorithm.

For each problem, several levels of accuracy were obtained by using the NESSUS code and the FPI algorithm. As a first step, a mean-based perturbation database was generated to generate a linear response surface. The result is called the mean-value-first-order (MVFO) solution.

In the second step, one or several probability levels were selected. For each probability level, the MVFO solution was then improved by replacing the center of perturbation (the "deterministic state" in the NESSUS/FEM module) by the most-probable points (design points) generated using the previously established linear response surface. The replacement of the deterministic value was accomplished by using the "MOVE" keyword in the NESSUS 2.0 code.

Table 6.1

STATUS OF PSAM VALIDATION CASES

CASE	DESCRIPTION	ANALYSIS TYPE	STATUS	SCHEDULE
1	Cantilever Beam	Static Correlated loading	Solution complete (Progress Report 87-3)	Complete
2	Cantilever Plate	Static Correlated loading	Solution complete (Progress Report 87-10)	Complete
3	Cantilever Beam	Natural frequency	Solution complete (Progress Report 87-7)	Complete
4	Rotating Beam	Centrifugal loading + Stress stiffening	Analytical solution complete (See 2nd Annual Report) NESSUS solution required	*FY88
5	Rotating Plate	Centrifugal loading + Stress stiffening	Solution complete (Progress Report 88-1)	Complete
6	Twisted Cantilever Plate	Natural frequency	Solution complete (Progress Report 88-1)	Complete
7	Plate	Correlated loading (Multiple zones)	Solution complete (Progress Report 88-1)	Complete
8	Shell	Static	Analytical solution required NESSUS solution required	Oct. 1987
9	Cylindrical Shell	Static	Solution complete (Progress Report 87-13)	Complete
10	Notched Plate	Stress Concentration	Solution complete (Progress Report 87-11)	Completed
11	Shell	Buckling	Solution complete NESSUS solution required	*FY88
12	Beam	Random vibration (See book by ELISHAKOFF)	Analytical solution required NESSUS solution required	*FY88
13	Cylindrical Shell	Random vibration. Problem same as #12, except for cylindrical shell. (See paper by ELISHAKOFF, VAN ZANTEN and CRANDALL)	Analytical solution required NESSUS solution required	*FY88
14	Plate	Random pressure field (See paper by DYER)	Analytical solution required NESSUS solution required	*FY88

*NOTE: Problem No. 4,11-14 not solvable using NESSUS version 2.0 (July 1987)

The "new" deterministic solution was then paired with the "old" MVFO probability estimates to form the Advanced MVFO (AMVFO) solutions.

The probability estimates were further improved by using the perturbation solutions around the updated point. This procedure is called the "first iteration." The solution can be further improved by using additional iterations until the solution (probability level, response value, or most probable point) converges. However, in all the validation problems studied, it was found that, from a practical point of view, the first iteration solutions were sufficiently accurate. In fact, it was found that even the AMVFO solutions provided good accuracy for most cases. Therefore, additional iterations were not conducted. The NESSUS probabilistic analysis algorithm are described in Refs. [2-4].

In solving the validation problems, user involvement was necessary to integrate the NESSUS/PRE, the NESSUS/FEM and the NESSUS/FPI modules. This slowed down considerably the solution process. However, based on the experience gained through the validation studies, an automated procedure has now being defined to be included in the PFEM module (see Chapter 4.0). It is anticipated that the user interactions in finding the probability solutions will be reduced considerably. The validation experience also has helped to design potentially more effective iteration algorithms as described in Chapter 3.0.

6.2 Validation Results Completed in FY '87

6.2.1 Static Analysis of Cantilever Beam (Case 1)

The exact solution for the validation problem 1 was included in the First Year Annual Report. The problem addressed is a cantilever beam subjected to static, statistically correlated point loads (see Figure A-1 in Appendix A). Other random variables include Young's modules, length, thickness, width, base spring and yield strength. The response function tested was the tip displacement.

The finite element model consisted of 20 Timoshenko beam elements. The NESSUS "mean" solutions of the tip displacement (0.3969 inches) agreed with the theory (0.4032 inches) within 1.5 percent. In this problem, the random variables were correlated. Therefore, the first step required that the NESSUS/PRE module be used to transform the correlated loads to uncorrelated random variables.

In solving this problem, it was found that the perturbation range for the length of the beam must be very small to avoid convergence instability. The perturbations used were 0.001 standard deviation for the length, and 0.1 standard deviations for the remaining variables.

The probability solution was checked by selecting three points in the right tail of the distribution (i.e., cumulative probability > 50%). The MVFO, the AMVFO, and the first iteration solutions are shown in Figure A-2 in Appendix A. The "exact" distribution shown in the figure was generated using Monte Carlo simulations with a sample size of 100,000.

Because there is a difference between the NESSUS/FEM solution and the theoretical solution, a "calibrated" or "adjusted" distribution curve was also established by matching the two (NESSUS and theoretical) solutions at the mean solutions. The adjusted curve provides a more reasonable reference to judge the accuracy of the NESSUS probabilistic solution.

By comparing the FPI solution with the adjusted solution shown in Figure V1-3, it can be concluded that the AMVFO and the first iteration solutions provide excellent probability estimates.

The result of this validation problem also suggests that the "small" numerical inaccuracy in the finite element solution (1.5% in the problem) may result in significant differences in the probability estimates. These differences may exceed the errors introduced by neglecting the second-order terms in the FPI algorithm. In other words, the first-order (i.e., using the response surface linearized about the design point) FPI method may be sufficient for practical applications. Nevertheless, the NESSUS code has the capability of dealing with second-order effect by generating more perturbation solutions and using quadratic response surfaces.

6.2.2 Static Analysis of Cantilever Plate (Case 2)

This validation problem is similar to case 1 except that the cantilever beam is changed to cantilever plate. To produce a reasonable model, the thickness of the beam as well as the magnitude of the loads were reduced. The response functions considered are the bending stress at the base and the tip displacement.

The finite element model consisted of 20 shell elements with 42 nodes as shown in Figure A-3 in Appendix A. The NESSUS "mean" solutions were 0.7648 inches for the displacement and 3657 psi for the stress. These values agreed with theory - 0.7692 inches and 3600 psi, respectively. The differences are 0.5% for the displacement and 1.6% for the stress.

For either the displacement or the stress, the probabilistic solutions were checked by selecting two points in the right tail of the distribution (i.e., cumulative probability > 50%).

In solving the problem, it was found that the perturbation range for the length and the width of the cantilever plate must be small (0.01 standard deviations for the length and the width, and 0.1 standard deviations for the remaining variables) to avoid convergence instability.

The MVFO, the AMVFO, and the first iteration solutions for the displacement and the stress, respectively, are shown in Figures A-4 and A-5 in Appendix A. The "exact" solution shown in the figures was generated by applying Monte Carlo simulation (sample size = 100,000) to the theoretical solutions.

Because the "small" difference in the stress values between NESSUS and the theoretical solution resulted in significantly different probability estimates, a "calibrated" stress distribution curve was established for judging the FPI solution algorithm. By comparing the NESSUS solutions with the adjusted solutions, it can be concluded that the AMVFO and the first iteration solutions provide excellent probability estimates.

6.2.3 Eigenvalue Analysis of Cantilever Beam (Case 3)

The goal of the validation problem 3 was to validate the NESSUS eigenvalue solution algorithms. The problem consisted of a cantilever beam. The response functions of interest were the first three bending frequencies in each of the two lateral directions. Exact CDF solutions are available for this problem (see PSAM 2nd Annual Report).

The random variables selected were: modulus, density, length, width and thickness. The mean thickness (0.98 in.) and the mean width (1 in.) were chosen to be approximately equal to test the ability of the code for identifying closely spaced eigenvalues.

The finite element model consisted of 20 beam elements (NESSUS element Type 98). The NESSUS "mean" solutions of the first six vibration modes were found to be in good agreement with the theory (neglecting the effects of rotary inertia and shearing deformations), with differences ranging from 0.2% to 2.2%. The accuracy of the perturbation results was judged by computing the sensitivities of the frequencies with respect to the perturbed random variables. It was found that the maximum error in sensitivities was 6.3% (for width perturbation). A summary of the NESSUS perturbation analysis is given in Table A-1 of Appendix A.

In solving the problem, it was found that the perturbation range for the length of the beam must be very small (0.001 standard deviation for the length and 0.1 standard deviation for the remaining variables) to avoid convergence instability.

Probability analysis results were generated for the first-mode solution only. However, the results for other modes are expected to have the similar accuracy based on the fact that the NESSUS generated sensitivities are accurate. Figure A-7 in Appendix A shows excellent agreement between the exact and the NESSUS solutions.

6.2.4 Eigenvalue Analysis of Rotating Beam (Case 5)

Validation problem 5 considers a rotating beam as illustrated in Figure A-8. There are five random variables: mass density, length, Young's modulus, thickness and width. This problem tests the centrifugal loading and stress stiffening capabilities in the NESSUS beam element. The response functions consider the tip axial displacement and the first bending frequency. The approximate frequency solution was derived by assuming a bending mode shape.

In the original test plan, the beam was fixed at the center of rotation. To represent a turbine blade configuration more closely, the inner radius (measured from the center of rotation to the "fixed" end of the beam) was defined to be 4.237 inches. Analytical solutions were revised and used to generate exact solutions using Monte Carlo simulation (sample size 500,000).

In solving the problem, it was found that the perturbation range for the length of the beam must be very "small" (0.001 standard deviation for the length) to obtain the correct perturbation solution. When the perturbation range was 1.0 standard deviation, there was no solution (convergence instability problem) and when the range was 0.1 standard deviation the generated perturbation result was incorrect, the frequency increased as the length increased. This perturbation problem is being investigated. All the key parameters for the eigenvalue perturbation are included in Appendix A for further testing.

Using the "small" perturbation range for the length, the probability analysis results were generated. Figures A-9 and A-10 in Appendix A show very good agreement between the "adjusted exact" solutions and the NESSUS AMVFO solutions.

6.2.5 Eigenvalue Analysis of Twisted Plate (Case 6)

The problem definitions and the solution are summarized in Appendix A. The geometry of the twisted plate was selected the same way as one of the test samples described in the paper by Macbain, Kielb and Leissa entitled, "Vibrations of Twisted Cantilever Plates - Experimental Investigation." The selected response functions were the first bending and the first torsion frequency.

A total of 192 shell elements (Type 75 - four-node shell) were used. The deterministic NESSUS solutions for the selected frequencies agreed well with the experimental results (about 4% difference). However, because the general theoretical solution for the twisted plate is unavailable for bending, torsion and mixed vibration modes, the validation of the probabilistic solution is only partially completed.

The "exact" solution for the first bending mode was based on the flat plate solution. For the selected geometry, this solution is reasonable based on the experimental results which suggests that the analytical solution can be used to predict, with good accuracy, the frequencies for different thicknesses. The probabilistic analysis solution using the advanced mean-value-first-order method (AMVFO) as shown in Figure A-13 agrees very well with the calibrated exact solution (adjusted so that the mean value FEM solution equals the experimental data). For the torsional mode, it was found that the flat plate solution cannot be used reliably to predict the results of the experiment. However, the probabilistic solution was obtained (Figure A-14) and can be used to compare with the theoretical solution should it become available.

6.2.6 Static Analysis Flat Plate (Case 7)

The problem definitions and the solution are summarized in Appendix A. The special feature of this problem is that the loads are applied to multiple "zones" as illustrated in Figure A-15. In each zone, the loads are either independent, partially correlated, or fully correlated.

The MVFO, AMVFO, and the first iteration solutions for the displacement are shown in Figure A-16 in Appendix A. The "exact" solution shown in the figure was generated by applying the Monte Carlo simulations (sample size = 500,000) to the theoretical solutions.

An adjusted exact stress distribution curve was established for judging the accuracy of the NESSUS solution. By comparing the NESSUS

solutions with the adjusted solutions, it can be concluded that the AMVFO and the first iteration solutions provide excellent probability estimates.

6.2.7 Static Analysis of Cylindrical Shell (Case 9)

This problem is a cylindrical shell subjected to axisymmetric ring loads. Seven random variables consisting of Young's modulus, inside radius, and five correlated loads were selected. The finite element used was NESSUS element Type 153 - a four-node assumed strain axisymmetric element. The finite element model had a total of 50 elements, and the element mesh is shown in Figure A-17.

The "exact" probabilistic solution was solved by using Monte Carlo simulation (sample size 500,000) with the theoretical solution taken from Timoshenko's "Theory of Plates and Shells." The difference between the deterministic (based on mean values) NESSUS and Timoshenko solution was 2.2 percent for radial displacement under the load.

Validation results for both the NESSUS/FEM code and the probabilistic analysis algorithm (FPI) were obtained (see Figures A-18 and A-19). Note that the validation of the NESSUS/FEM code was based on the FEM solution, and the validation of the FPI algorithm was based on the Timoshenko solution. The probabilistic analysis procedure, however, is identical for both solutions.

The perturbation range was chosen as 0.1 standard deviation for each random variable. It was found that the NESSUS/FEM solution required very tight convergence limits for generating accurate Young's modulus sensitivity data. Also, it was found that this convergence problem can be solved by increasing the perturbation range to 0.5 standard deviations.

Figure A-18 and Figure A-19 present the MVFO and the AMVFO solutions. If required, accuracy can be improved by applying the iteration procedure. However, Figure A-19 indicates that the AMVFO solution is sufficiently accurate for this problem. Therefore, no iteration solution was obtained for NESSUS/FEM.

For the NESSUS/FEM solution (Figure A-18), a calibrated "exact" solution was again used to compare with the NESSUS/FEM solution. Figure A-19 shows that NESSUS solutions and adjusted solutions are very close, thus, indicating that the AMVFO solution provides very good probability estimates.

6.2.8 Analysis for Stress Concentration Factor (Case 10)

The response function considered was the maximum stress at the notch of an axially loaded sheet in Figure A-201. The radius of the notch is defined as the random variable which has a lognormal or truncated-normal distribution. The problem definitions and the solutions are summarized in Appendix A. The radius is not a standard input in NESSUS/FEM, however, this validation case shows that the user can define a geometry parameter as a random variable by providing proper perturbed coordinate data in the NESSUS/FEM random variables setting.

Because the response is a function of one random variable, it can be shown that, theoretically, the advanced MVFO method should yield the exact CDF solution. Therefore, the difference between the NESSUS solution and the exact solution (Figure A-21) is due to the error in the finite element solution. However, the error is small (about 1% in stress). Note that, for the case where the radius has a truncated distribution, the resulting NESSUS probability distribution is also truncated (Figure A-22), as expected.

6.3 Validation Plans for FY '88

6.3.1 Summary of FY '88 Effort

The validation cases planned are listed in Table 6.1. The emphasis will be on dynamic problems and response to random loading. Descriptions of the planned validation problems follow.

6.3.2 Eigenvalue Analysis of Rotating Beam (Case 4)

Validation case 4 is the same as case 5 except that the finite elements are Timoshenko beam elements.

6.3.3 Static Analysis of Shell (Case 8)

Validation case 8 is a static problem. The main goal is to validate the general two-dimensional shell (non-axisymmetric) element in the NESSUS library.

6.3.4 Buckling Analysis of Cylindrical Shell (Case 11)

The random variables will be shell thickness and the applied pressure. This problem has been solved using the "move" option. However, the solution using the NESSUS perturbation scheme has not been obtained using the NESSUS 2.0 code.

6.3.5 Random Vibration Analysis of Beam (Case 12)

This problem was described in detail in [1]. A concentrated random loading defined using a power spectral density function is applied to a

simply-supported beam. The response of interest is the displacement, and approximate solutions for the mean and standard deviation of the response are available.

6.3.6 Random Vibration Analysis of Cylindrical Shell (Case 13)

This problem is similar to case 11. The structure is a cylindrical shell subjected to a random uniform ring loading at a section of the shell [5].

6.3.7 Random Pressure Loads on Plate (Case 14)

The goal of this validation case is to validate NESSUS' capability to solve random pressure field problems. In this validation case, a plate is subjected to a random pressure field [6].

References

[1] "Probabilistic Structural Analysis Methods (PSAM)," 1st Annual Report, NASA Contract NAS3-24389, Vol. III, Section 5.

[2] "Probabilistic Structural Analysis Methods (PSAM)," 2nd Annual Report, NASA Contract NAS3-24389, Appendix F.

[3] T. A. Cruse, O. H. Burnside, Y.-T. Wu, E. Z. Polch, P. K. Fink, and J. B. Dias, "Probabilistic Structural Analysis Methods for Select Space Propulsion System Structural Components," submitted to Computers & Structures, September 1987 (PSAM).

[4] Y.-T. Wu, O. H. Burnside and J. Dominguez, "Efficient Probabilistic Fracture Mechanics Analysis," Fourth International Conference on Numerical Methods in Fracture Mechanics, San Antonio, Texas, March 1987.

[5] I. Elishakoff, A. Th. van Zanten and S. H. Crandall, "Wide-Band Random Axisymmetric Vibration of Cylindrical Shells," Journal of Applied Mechanics, Vol. 46, June 1979.

[6] I. Dyer, "Response of Plates to a Decaying and Convecting Random Pressure Field," The Journal of the Acoustical Society of America, Vol. 31, No. 7, July 1959.

7.0 NESSUS CODE VERIFICATION STUDY

7.1 Scope of Verification Problems

The purpose of the verification efforts is to apply the Probabilistic Structural Analysis Methods (PSAM) to the analysis of actual typical aerospace propulsion system components. Four components, typical of the hardware found in rocket propulsion engine systems have been chosen for this application. They are turbine blade, high pressure duct, LOX post and transfer tube liner. These components are subject to environments with many random variables. Detailed discussion of the environments, failure modes and the deterministic analysis techniques were reported as part of the first annual report.

A wide range of probabilistic structural analysis tools will be or have already been implemented in the NESSUS/FEM code. The verification studies have been tailored such that different areas of structural mechanics are emphasized on each of the components. This has been done consistant with the primary design requirement for each component.

The turbine blade analysis concentrates on linear static and modal frequency extraction analysis. The duct application emphasizes the random vibration capabilities within the linear dynamics domain. The LOX post application involves the use of nonlinear material analysis. The transfer tube liner application involves material and geometric nonlinear analysis. All the efforts on the above components analyse various response variables in the probabilistic domain.

Initial verification efforts concentrated on the accuracy, robustness, and efficiency of the methodologies implemented in the NESSUS/FEM code. Several test cases were run using NESSUS/FEM and the results were compared with results from commercial codes such as ANSYS. The initial studies pointed the way to improvements in user interface, analysis tools, and element formulation. Some of the details of these studies can be found in the earlier annual and monthly reports.

7.2 Turbine Blade Random Variables

A high performance, high pressure fuel turbopump second stage blade was considered for this study. The blade is made of single crystal PW1480 material which has directional properties. The following variables have been

identified as random and will be considered for the probabilistic linear static analysis.

1. Material axis orientation
2. Single crystal material elastic constants
3. Geometry
4. Centrifugal load
5. Temperature load
6. Pressure load

The initial study considering the first three items is reported herein. The random load variables will be included in the subsequent effort; the contribution of loading can be analysed by adding the NESSUS/FEM results to the existing database.

Statistical data for material axis orientation were obtained from a set of approximately one hundred blades. For these single crystal blades the primary material axis was controlled but not the secondary axis orientation. The statistical analysis of the data indicated a standard deviation of 3.87 degrees for the primary axis orientation. Further, there was no correlation observed between the primary and secondary axis data. The new blades that will be manufactured and tested will have both the primary and secondary material axis controlled. This study considers the material axis orientations, both primary and secondary, as independent random variables each having a standard deviation similar to that observed in the set of one hundred blades discussed above. Analysis of data from a small sample of blades where primary and secondary axes were controlled indicate similar standard deviations. For the purposes of this study, a normal distribution was assumed.

The elastic material constants were assumed to be functions of temperature and were introduced through the use of user subroutines in NESSUS/FEM. The material properties used is reported in the Table 7.1. The variations in elastic constants in single crystal materials is considered to be small. A coefficient of variation of 0.025 was used for all the elastic constants. The standard deviation was assumed to be the same at all temperatures.

Table 7.1

Material Properties for the Turbine Blade Model

TEMP	E	NU	G	ALPHA
RANKINE	PSI		PSI	INCH/INCH/R
60	19.95E6	0.376	20.50E6	2.30E-6
360	18.82E6	0.382	19.30E6	3.80E-6
530	18.38E6	0.386	18.63E6	4.65E-6
660	17.61E6	0.389	18.00E6	5.29E-6
1860	14.79E6	0.395	15.27E6	7.76E-6
2060	13.91E6	0.401	14.60E6	8.07E-6

Mass density = 0.805E-3 $lbf.sec^2/in.^4$

The nature of the geometrical variations in a turbine blade shape is a function of the manufacturing methods. Procedures have been implemented in the NESSUS/EXPERT system to introduce many types of geometric perturbations to the finite element model. These include uniform volume increase or decrease, geometrical translation and/or rotations about some arbitrary set of axes. For cast and then machined blades such as the one being analysed in this study, actual measured data indicate that the majority of geometrical differences from blade to blade occur as rigid body shift and/or rotation about the stacking axis. Thus, geometrical perturbations as rigid body shifts of lean, tilt and twist angles have been introduced in this study. That is, the relative change of the center of mass (CG) with reference to stacking axis is more critical to stress analysis than the minor profile variation from blade to blade. Consequently, the lean, tilt, and twist angles have been treated as random variables (Figure 7.1). The data from a similar set of LOX blades was used to determine the standard deviations of these geometric angles. These three geometric angles were converted, through a preprocessor, into equivalent nodal coordinates and were then input to NESSUS.

In summary, a total of nine random variables were introduced in this first set of verification study. They are listed in Table 7.2. The study will be extended in the next phase to include the load random variables of speed, pressure and centrifugal load.

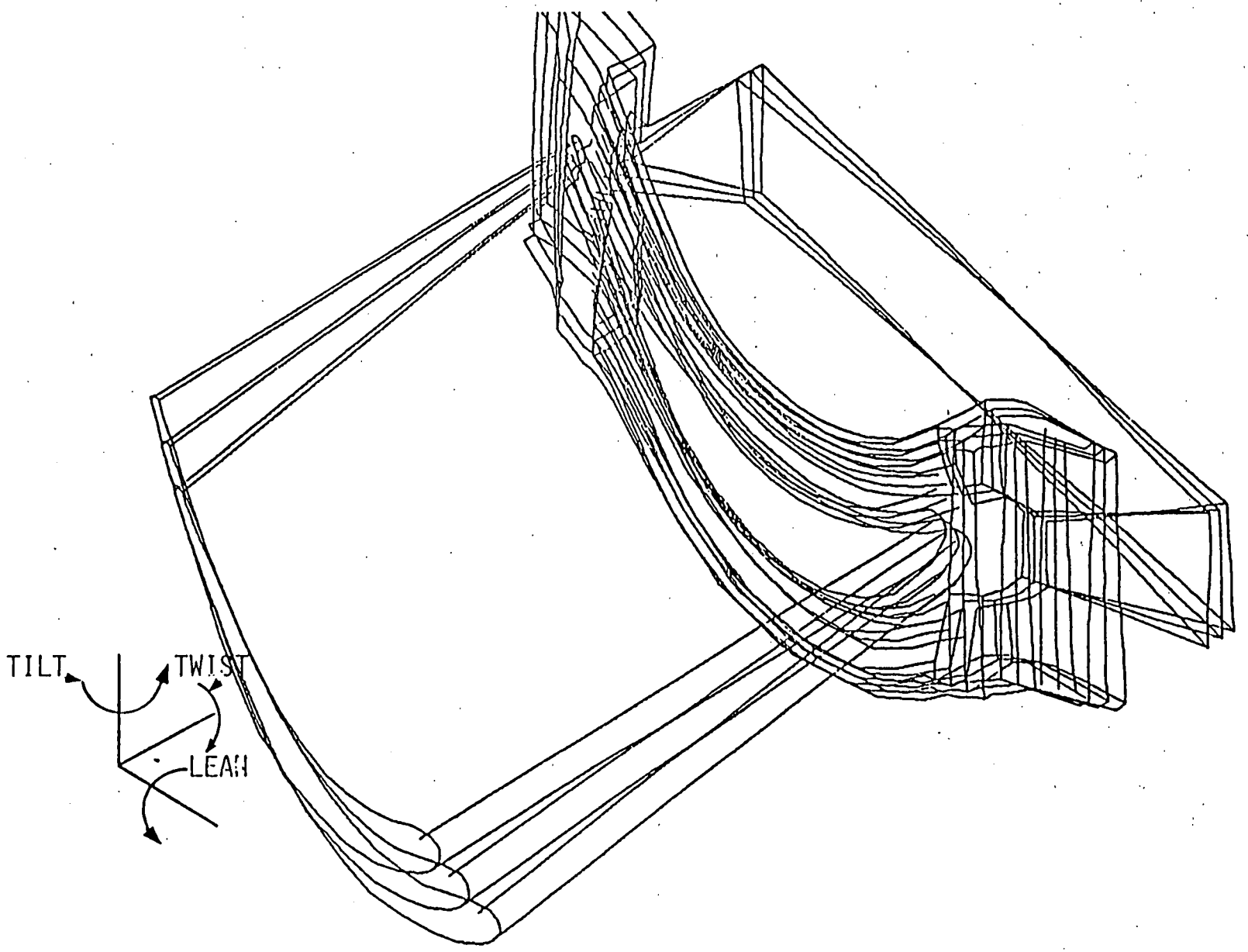


Figure 7.1 Geometric Perturbations Introduced in the Verification Study

Table 7.2

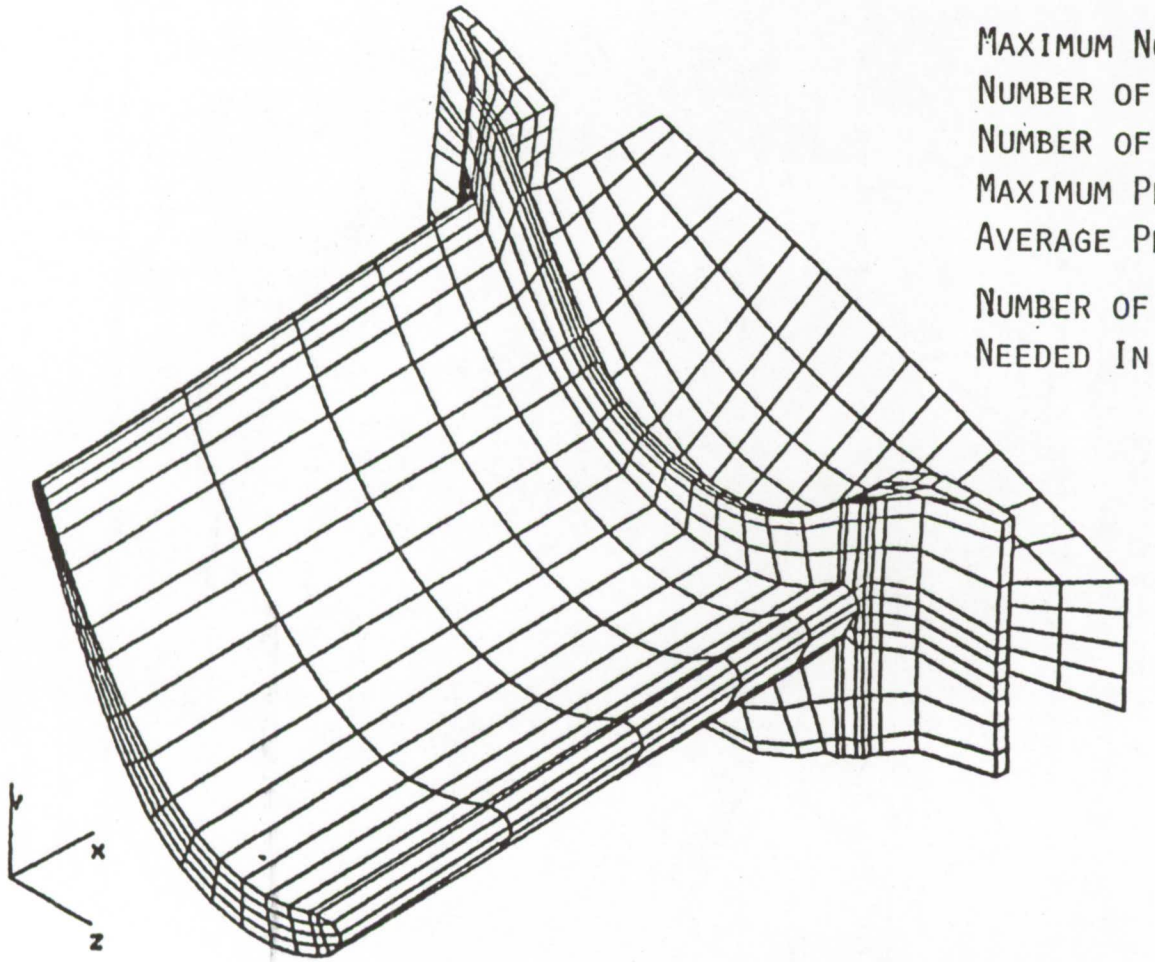
Random Variable Data for Blade Verification Study

Random Variable Number	Description	Mean Value	Std. Deviation	Distribution
1	Mat. orien. Theta Z	+0.05236 (rad)	0.067544(rad)	Normal
2	Mat. orien. Theta Y	-0.034907(rad)	0.067544(rad)	Normal
3	Mat. orien. Theta X	+0.082766(rad)	0.067544(rad)	Normal
4	E	Temp.Dependent	0.4596E6lbs/sq.inch	Normal
5	NU	Temp.Dependent	0.00965	Normal
6	G	Temp.Dependent	0.46575E6lbs/sq.inch	Normal
7	Geom.Lean	0.0 (degree)	0.14 (deg)	Normal
8	Geom.Tilt	0.0 (degree)	0.14 (deg)	Normal
9	Geom.Twist	0.0 (degree)	0.30 (deg)	Normal

7.3 Turbine Blade Verification Study Results

The finite element model used in this study is shown in Figure 7.2. The blade is subjected to complex pressure and temperature profiles shown in Figure 7.3 and Figure 7.4, respectively.

The probabilistic analysis results, considering the nine random variables discussed earlier, are presented below. The mean-value, first-order (MVFO) solution consists of one deterministic analysis (at the mean value state) followed by nine perturbation analyses, one for each random variable. The perturbation setting of 0.1 times the standard deviation was used to compute the gradients near the mean values. A NESSUS/POST FORTRAN interface program is available that will convert the geometry, displacements, stresses and strains available in the perturbation database into PATRAN readable, neutral and results files. The NESSUS/FPI module was modified to write the FPI results data into a PATRAN readable results file. Further, the NESSUS/FPI



MAXIMUM NODE NUMBERS = 2519
NUMBER OF ELEMENTS = 1456
NUMBER OF ACTIVE D.O.F. = 5946
MAXIMUM PROFILE HEIGHT = 578
AVERAGE PROFILE HEIGHT = 361

NUMBER OF 64 BIT WORDS = 2846215
NEEDED IN BLANK COMMON

Figure 7.2 Blade Verification Study Finite Element Model Details

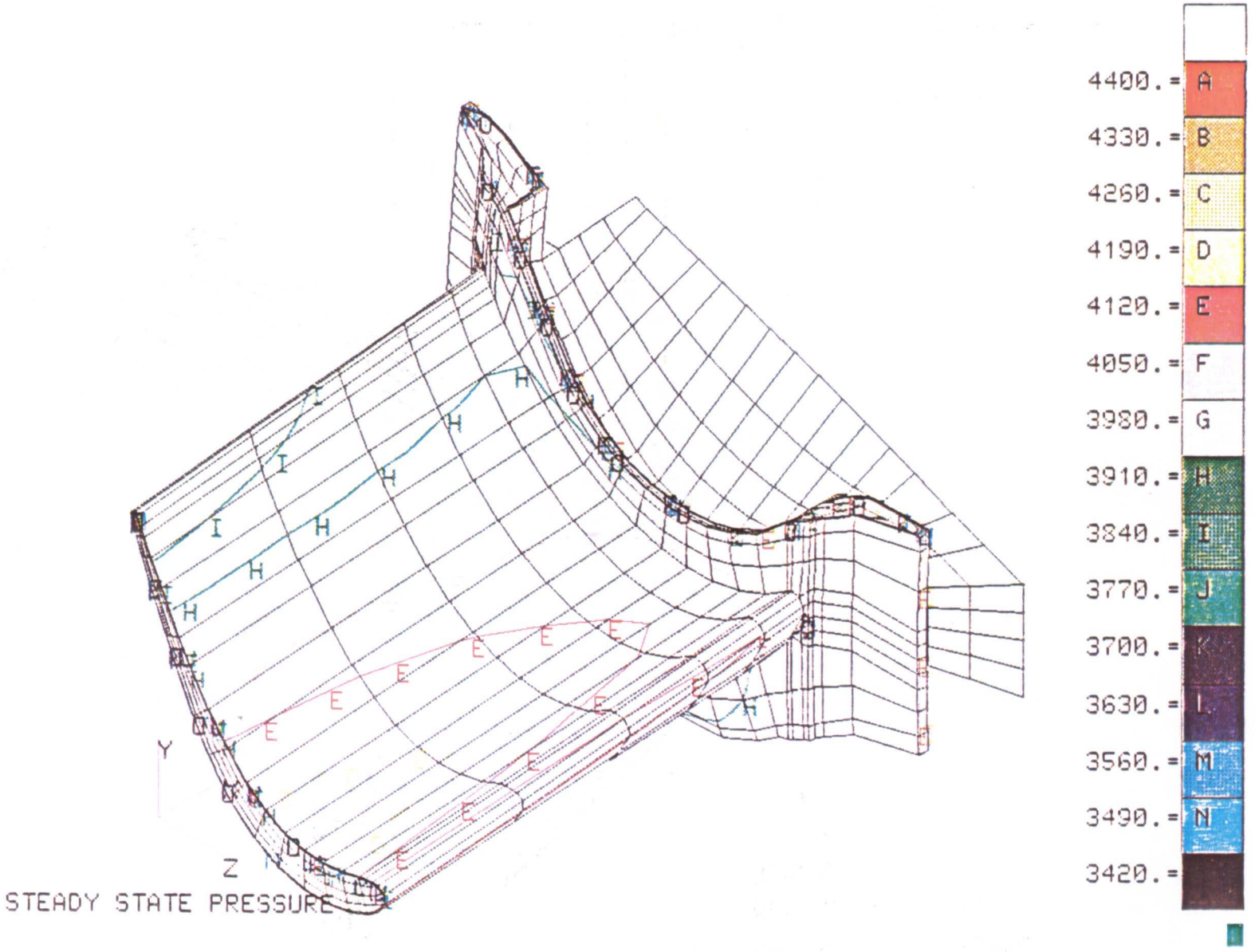


Figure 7.3 The Applied Steady State Pressure On The Finite Element Model

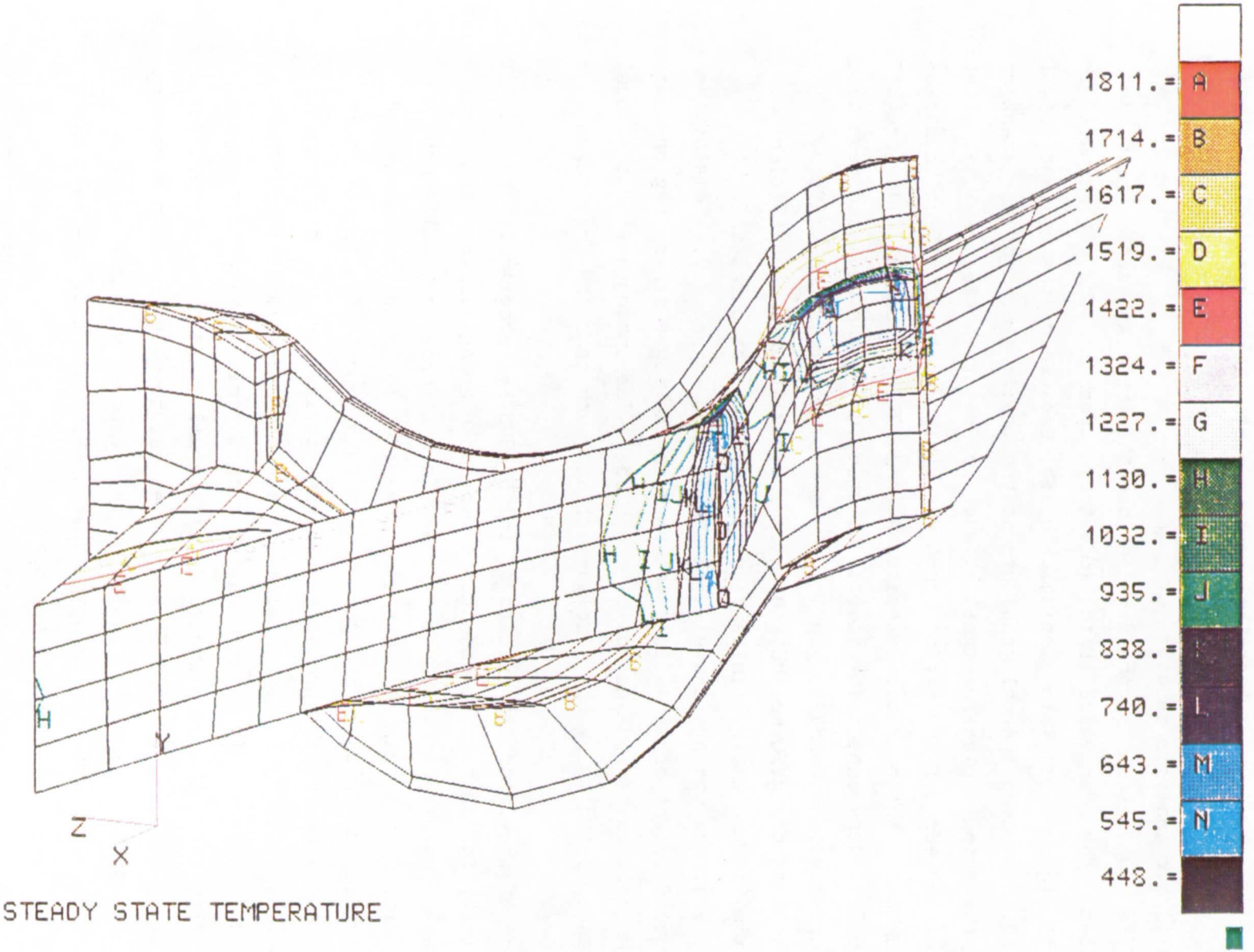


Figure 7.4 The Applied Steady State Temperature on the Finite Element Model

module was modified to process the entire model (2519 nodes) for a given response variable for the mean-value, first-order method. The results are presented in the form of graphical contour plots. These graphical plots aid in an effective interpretation of deterministic, perturbation as well as probabilistic analysis results.

Effective stress is considered as one the important stress response variables. The mean value of the effective stress for the entire model is shown in Figure 7.5. This particular blade, because of the coolant flow from the disk, has steep thermal gradients at the trailing edge of the shank root and at the shank - platform trailing edge intersection (Figure 7.4). Further, the trailing edge of the airfoil root has a high critical effective stress. Three nodes at their critical locations (node 2470, node 2518 and node 817) (Figure 7.4) were chosen for additional study using the advanced mean-value first-order method (ADMVFO), in which the design points are shifted.

Based on the MVFO method, the standard deviation and coefficient of variation for the entire finite element model were calculated for the effective stress and are shown in Figure 7.6 and Figure 7.7. Some of the larger coefficient of variations occurred in the lower stress regions away from the critical areas and inside the blade. The inaccuracy of the nodal stresses computed using the displacement approach near the free edges is also noted.

One of the important results of the NESSUS/FPI program is the relative sensitivity information of each random variable among all the random variables considered in that particular analysis. This information, called the sensitivity factor, can be plotted on the model for each random variable as shown in Figure 7.8 through Figure 7.16. This sensitivity factor, more appropriately called the probabilistic sensitivity factor, is a combination of physical sensitivity and uncertainty of the random variable measured by the standard deviation. In other words, a random variable with high physical sensitivity but with low standard deviation will have a low probabilistic sensitivity and vice-versa. This provides valuable information regarding the importance of each random variable for the response variable being considered. It might also be noted that the influence of the random variables differs in various regions of the blade.

In addition to the sensitivity information using the NESSUS/FPI and MVFO methods, one can map the probability of exceedence for the response variable

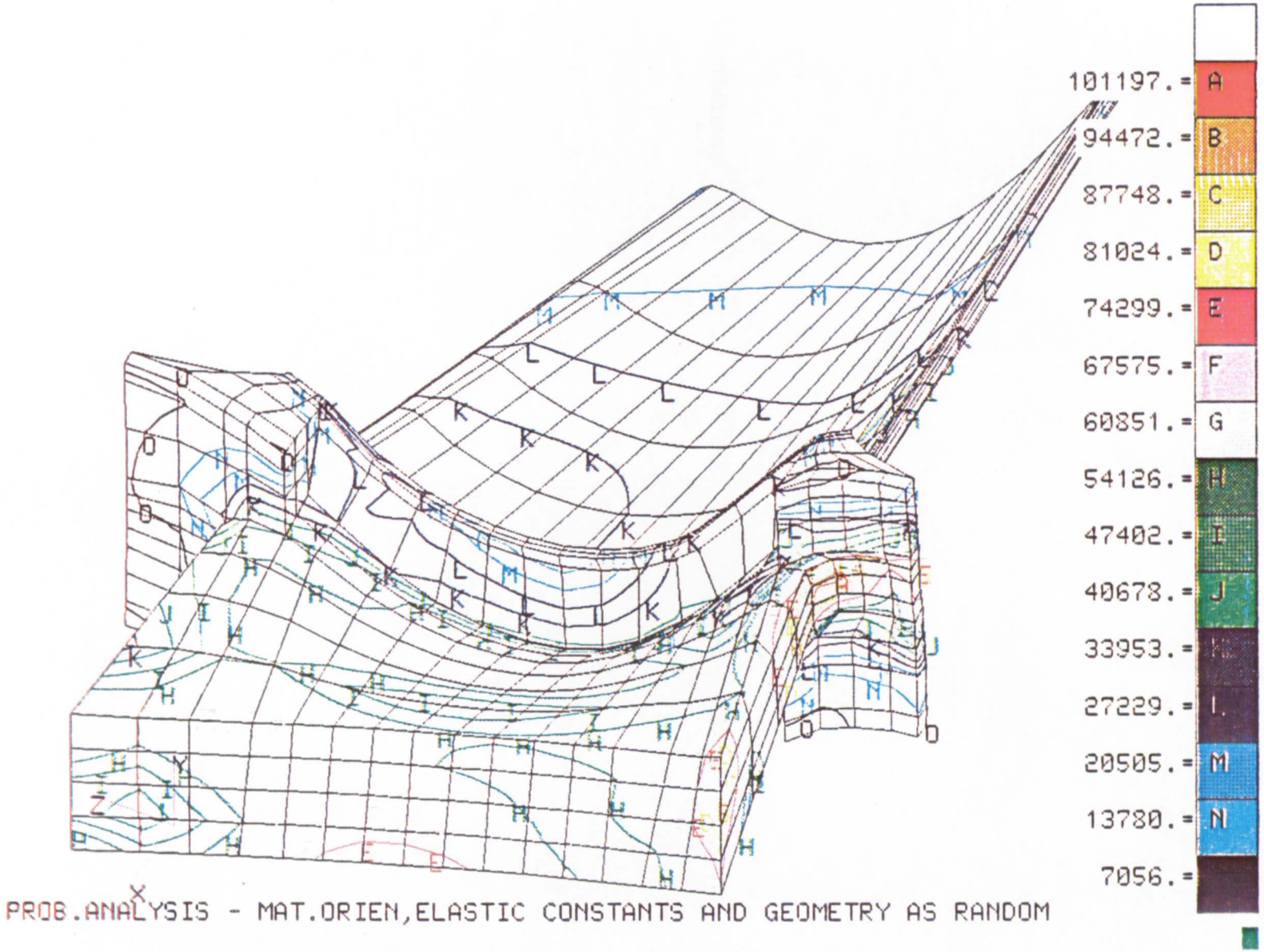


Figure 7.5 Effective Stress Mean Values

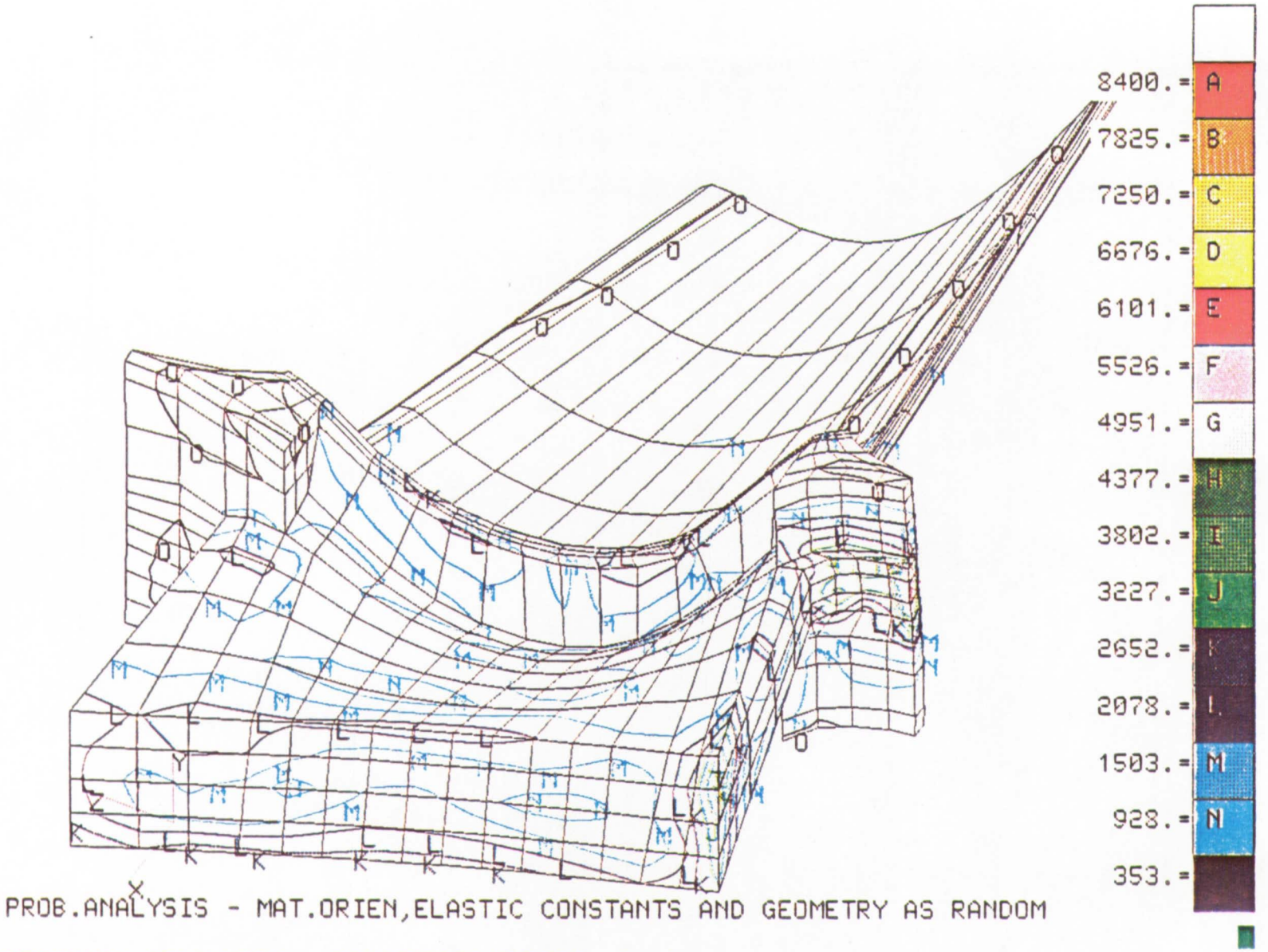


Figure 7.6 Effective Stress Standard Deviations

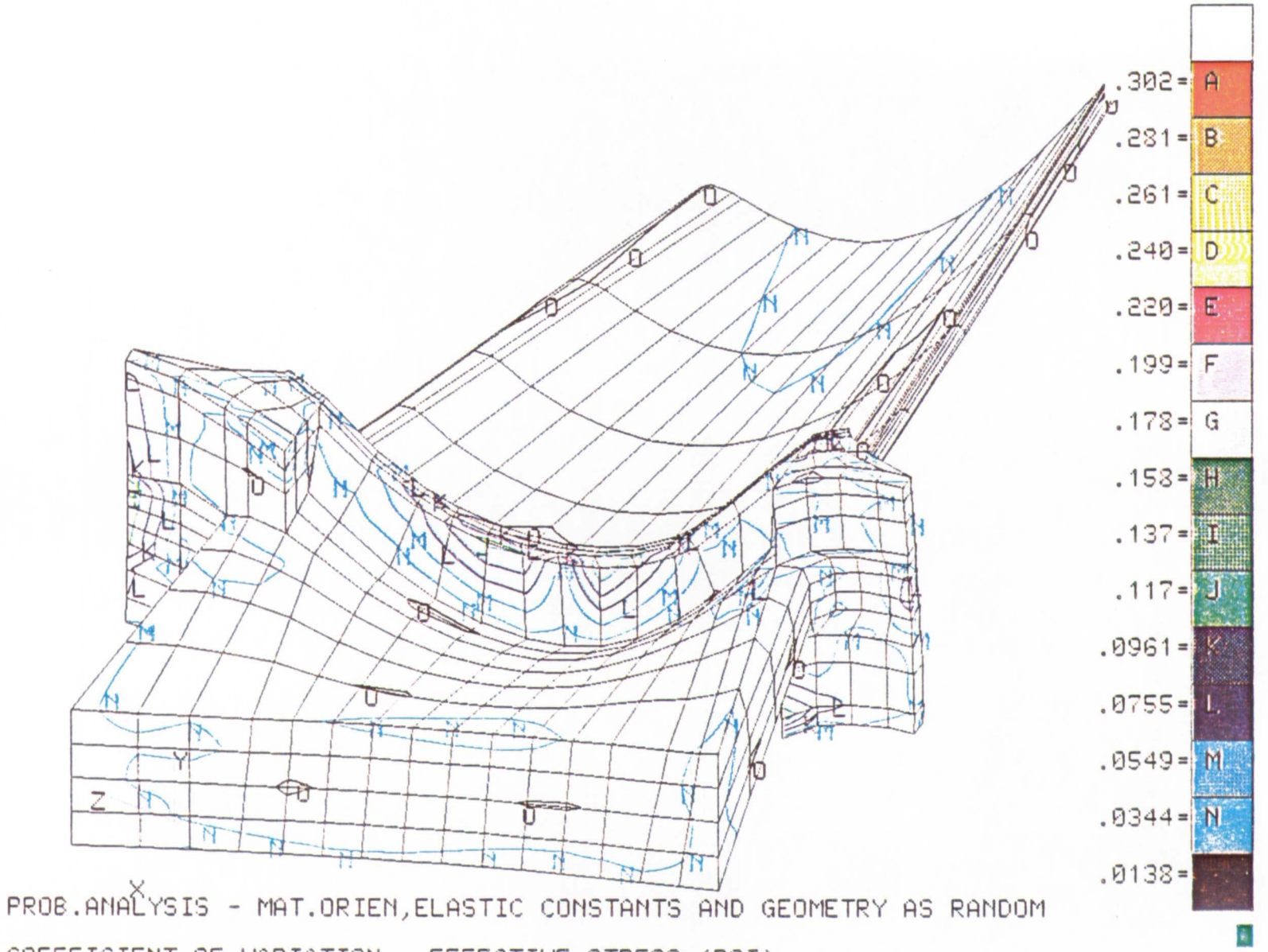


Figure 7.7 Effective Stress Coefficient of Variation

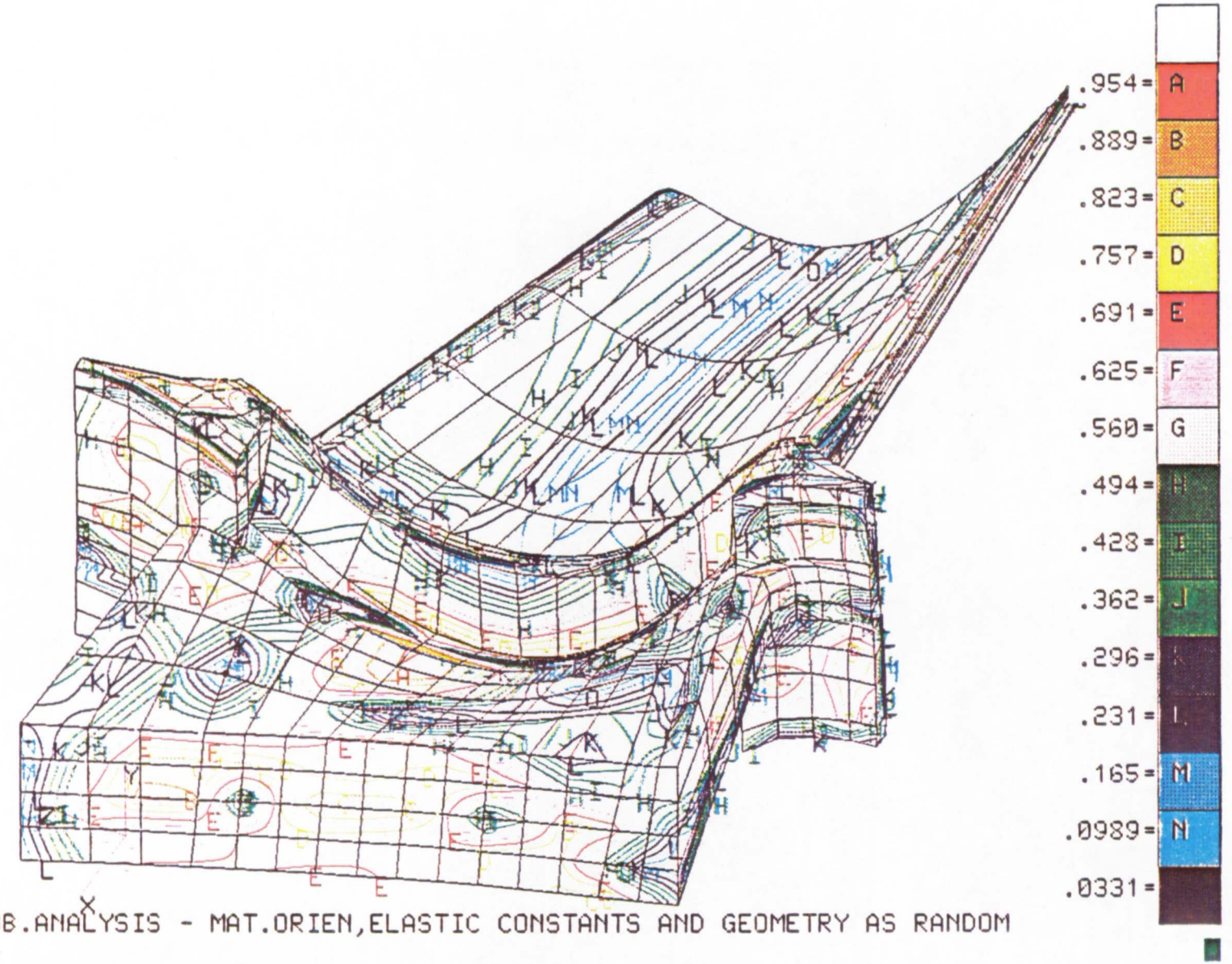


Figure 7.8 Effective Stress Sensitivity to Material Axis Orientation Lean Angle

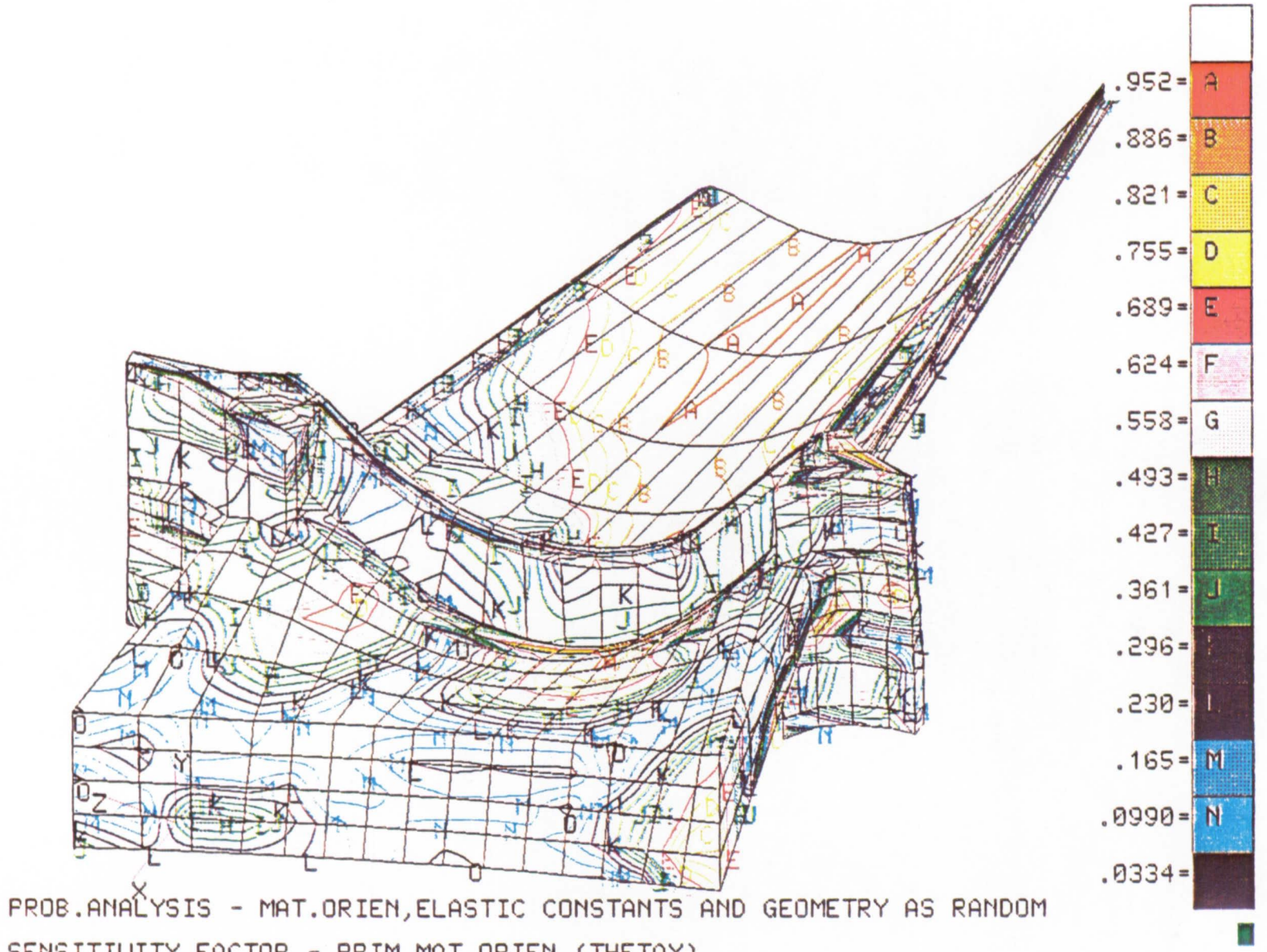


Figure 7.9 Effective Stress Sensitivity to Material Axis Tilt Angle Orientation

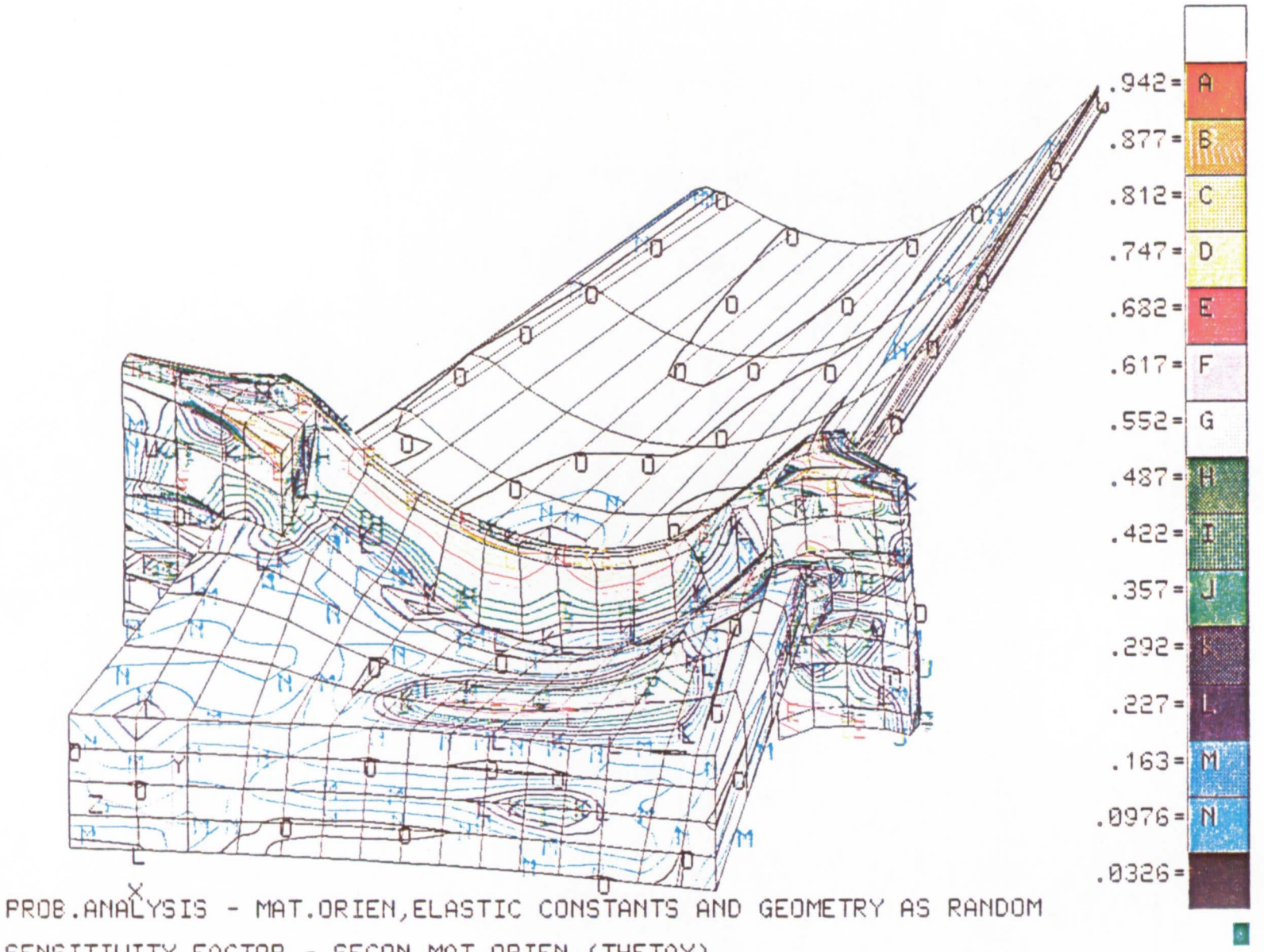


Figure 7.10 Effective Stress Sensitivity to Material Axis Twist Angle Orientation

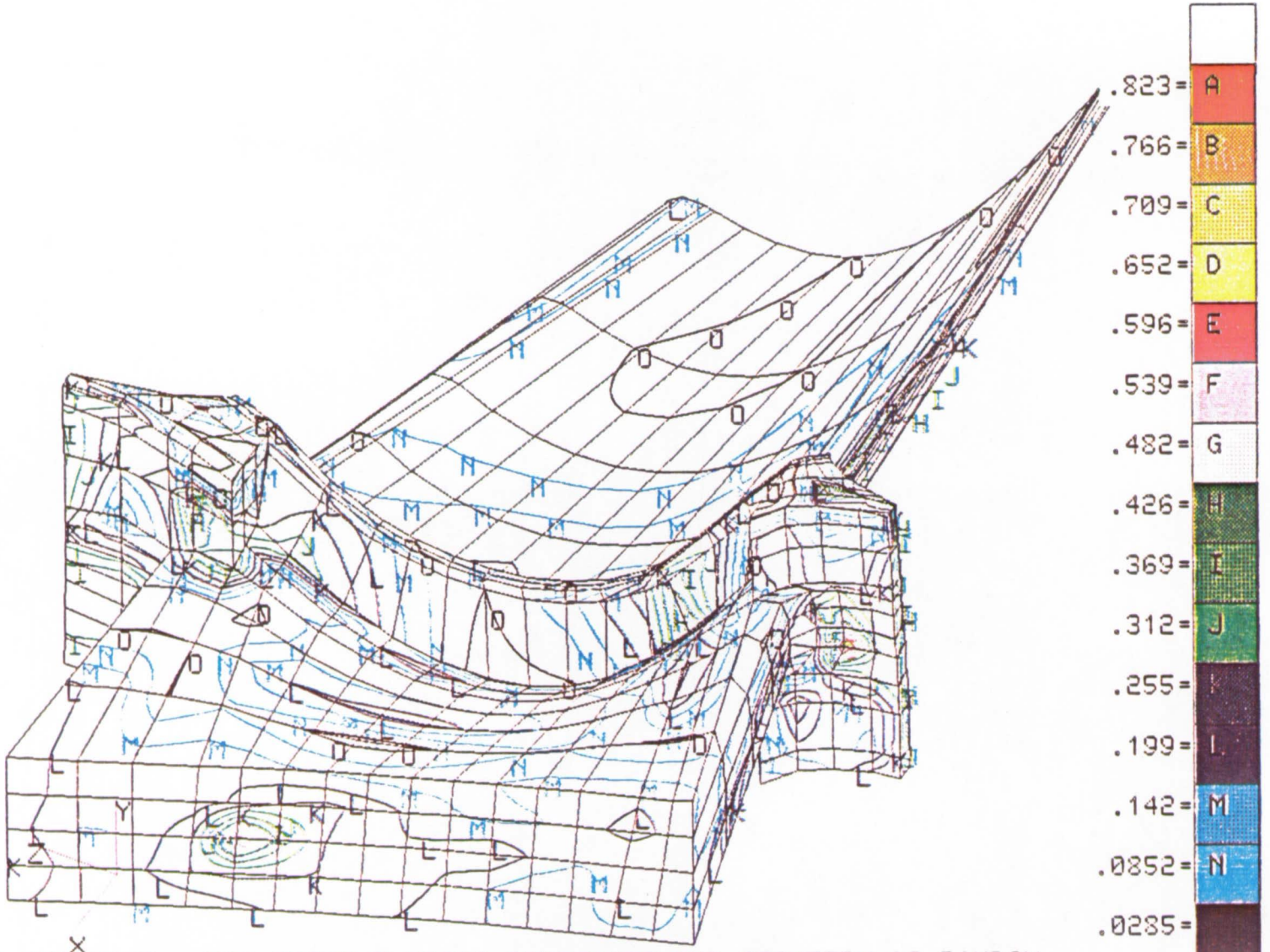


Figure 7.11 Effective Stress Sensitivity to Young's Modulus

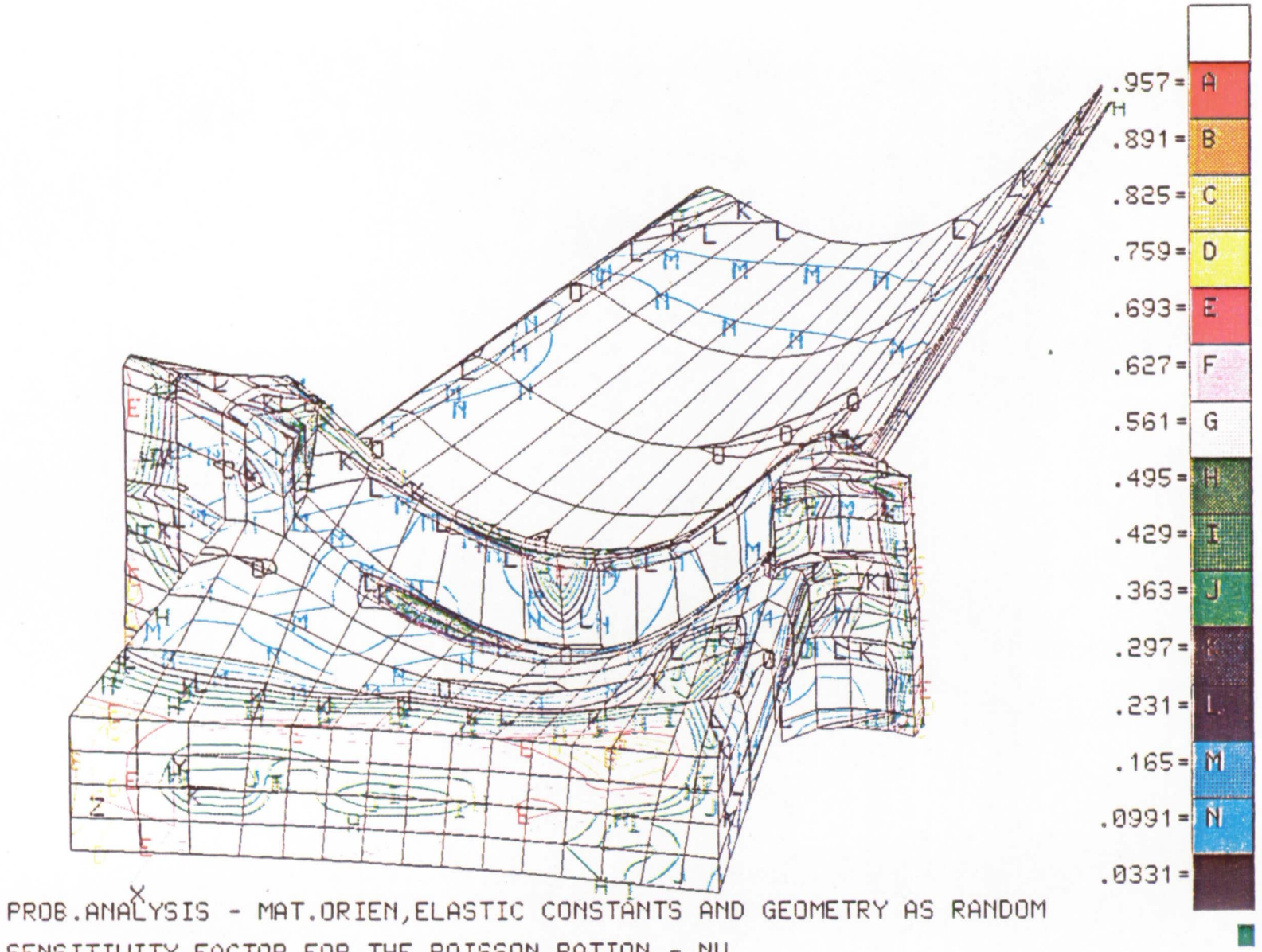


Figure 7.12 Effective Stress Sensitivity to Poisson's Ratio

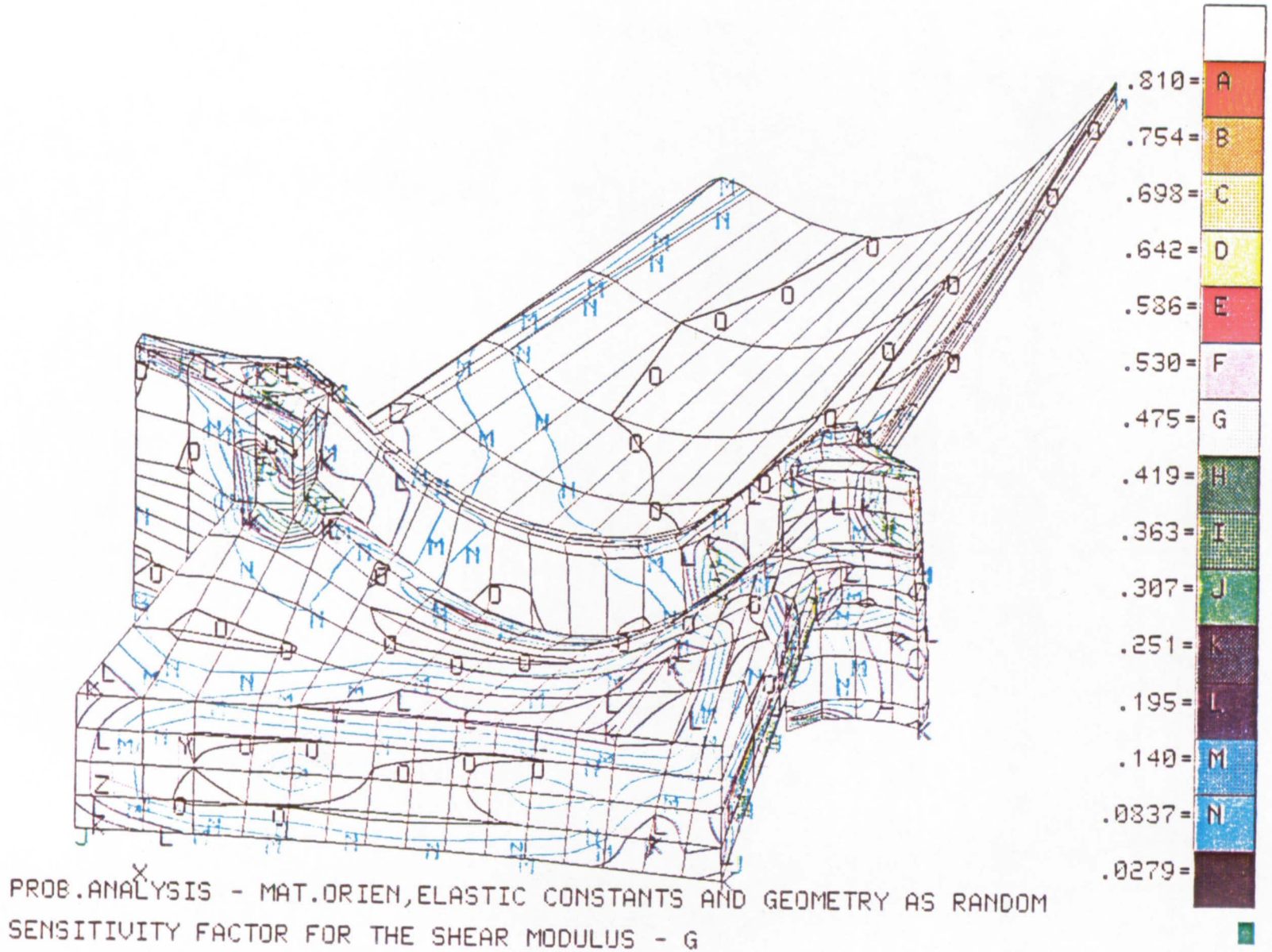


Figure 7.13 Effective Stress Sensitivity to Shear Modulus

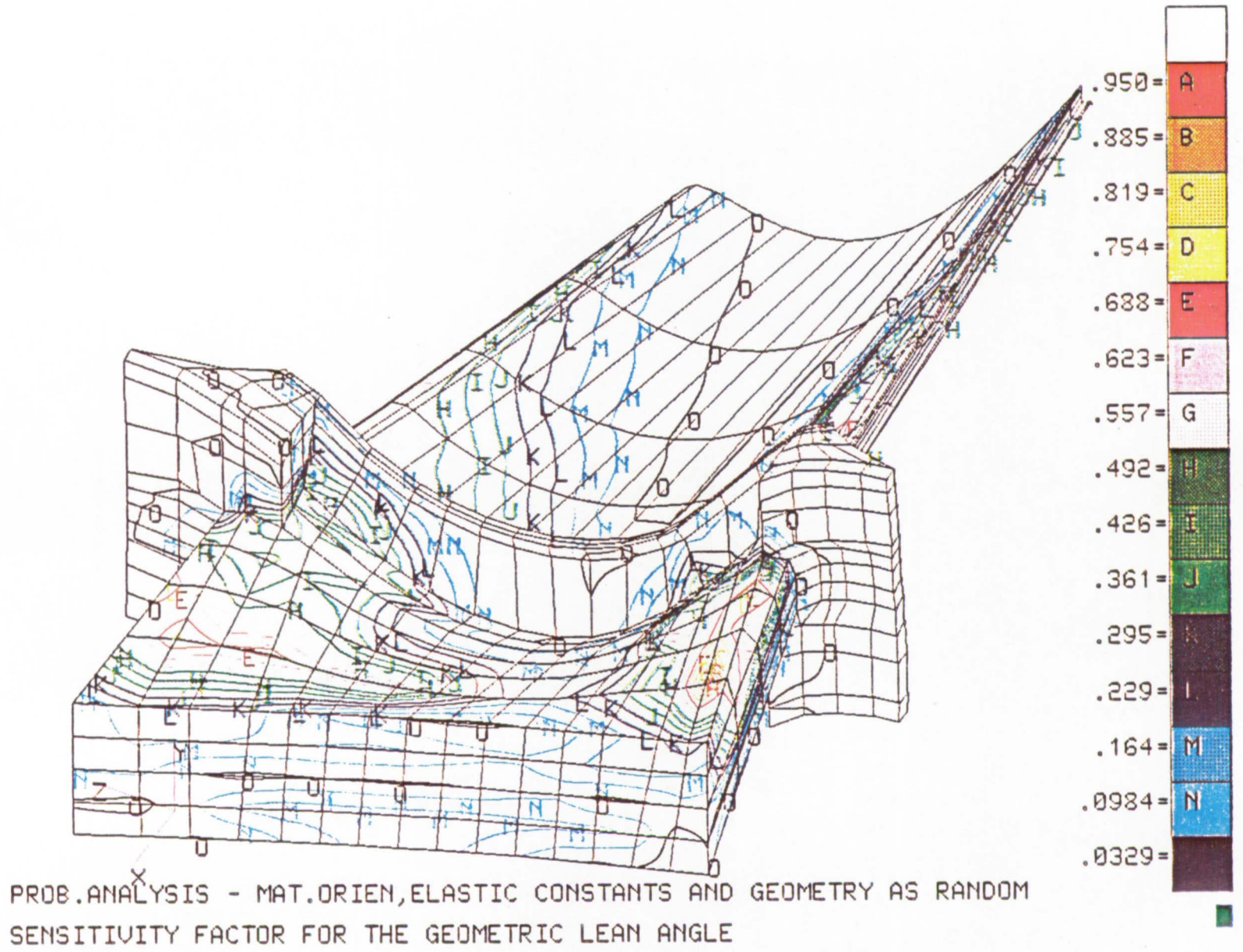


Figure 7.14 Effective Stress Sensitivity to Geometric Perturbation Lean Angle

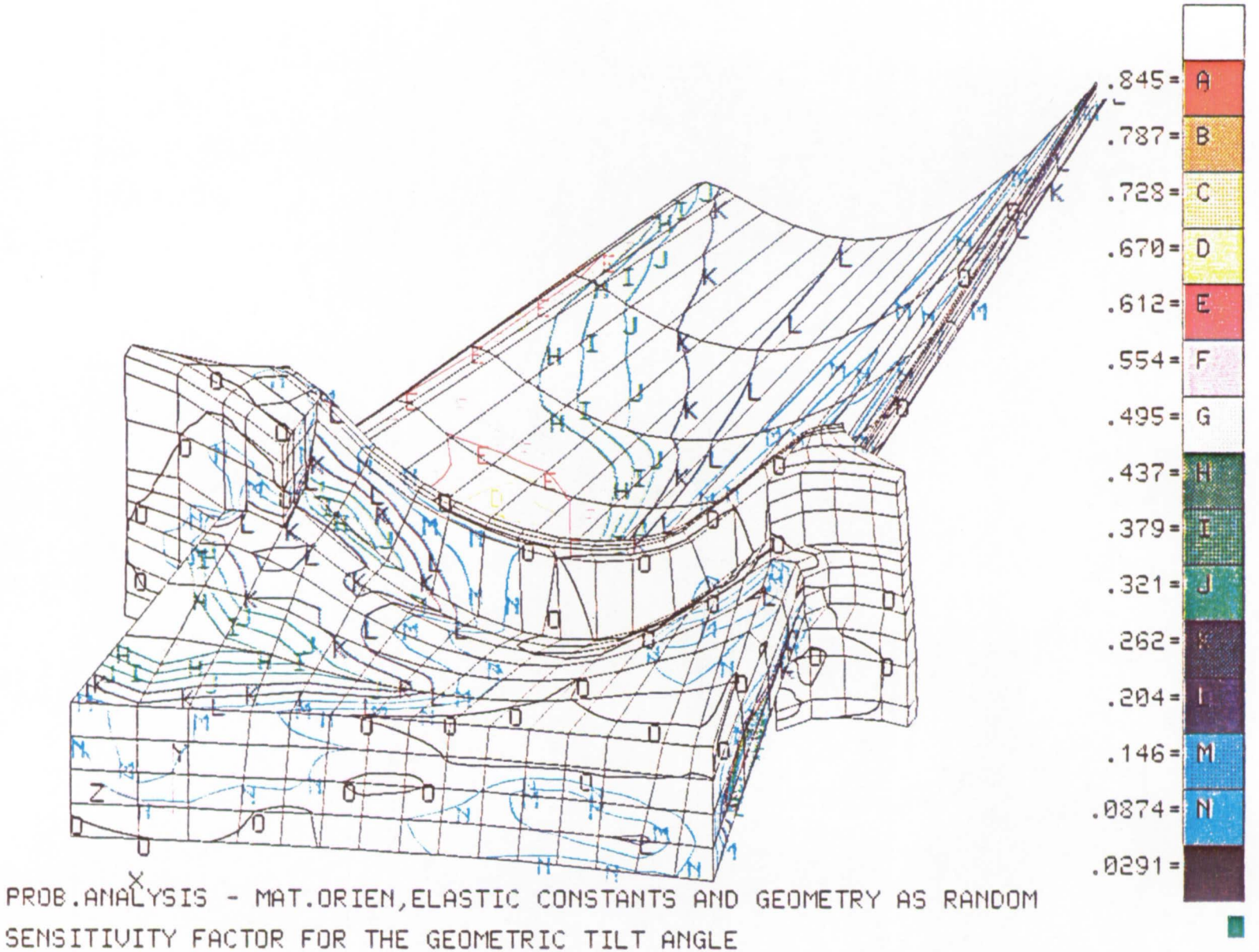


Figure 7.15 Effective Stress Sensitivity to Geometric Perturbation Tilt Angle

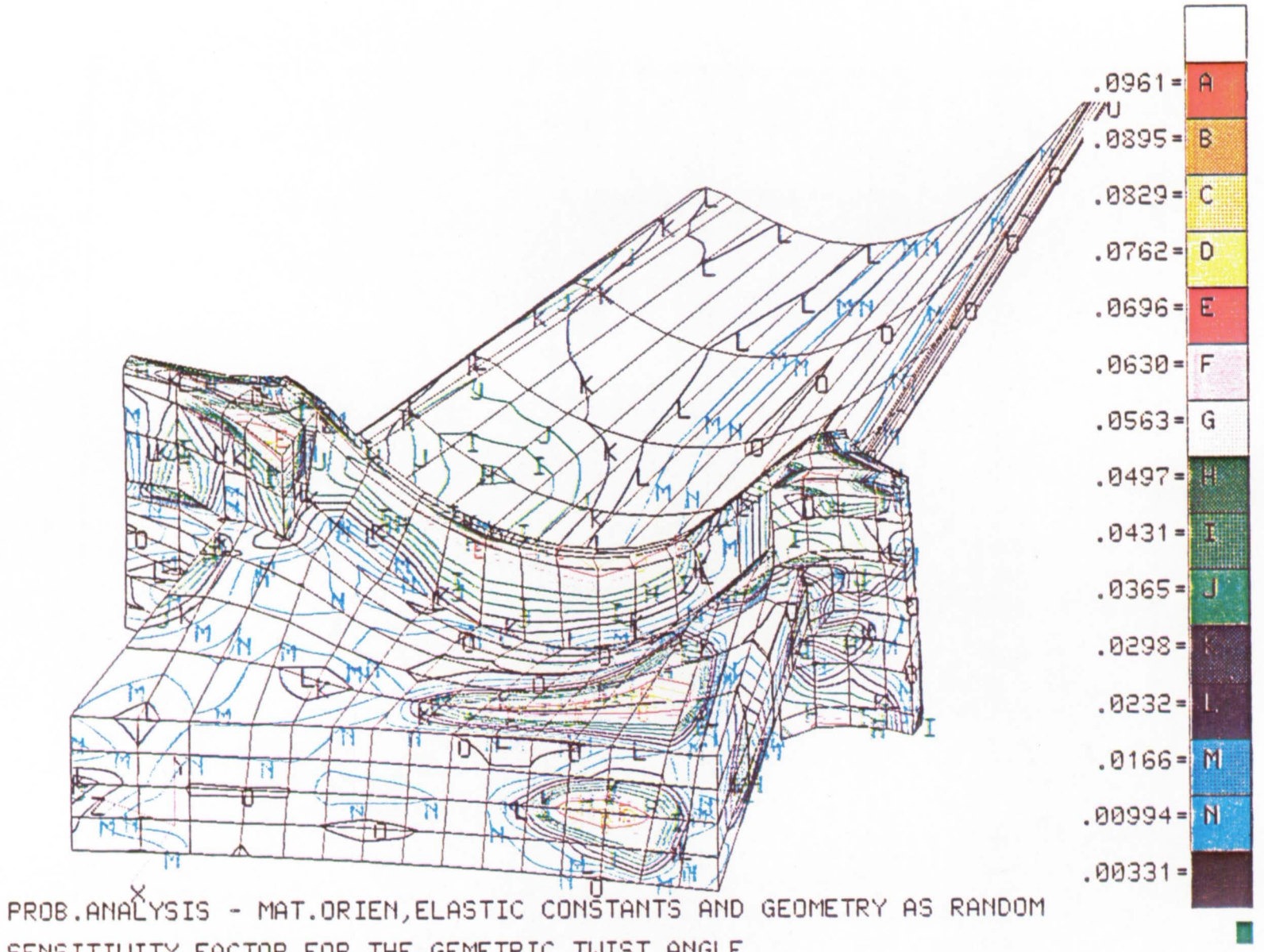


Figure 7.16 Effective Stress Sensitivity to Geometric Perturbation Twist Angle

for many different levels with one single run of NESSUS/FPI. This is illustrated in Figure 7.17 through 7.19 where the probability of exceedence of effective stress is plotted. This can be used to quickly identify critical areas of high stress and identify probable nonlinear material behavior regions.

After initial processing of the entire model for the effective stress, three nodes in the critical regions were selected for further processing. They are node 2470 and node 2518 in the shank root region and node 817 in the airfoil root region (Figure 7.4). The cumulative probability distribution functions based on MVFO method are shown in Figure 7.20 through Figure 7.22.

The probabilities for the effective stress are also represented in a different form in Figure 7.23 through Figure 7.25. For the advanced mean-value, first-order method, the finite element analysis was again run corresponding to three different levels of probability for the response variable: 1-sigma, 2-sigma, and 3-sigma from the mean. For each level, the deterministic solution was moved to the design point as calculated by NESSUS/FPI. The NESSUS/FPI was again used to successfully process this new deterministic data at the respective design points but using old perturbation data obtained around the mean values. The results are shown in Figure 7.23 through Figure 7.25 under the legend "ADMVFO" method. It is seen that for the nodes 2518 and 2470, the difference between the two methods is rather small indicating the linearity of the response function. However, at node 817, the differences between the two methods were significant enough to further process the results. At the 3-sigma level of the design point, perturbations were again calculated at node 817 for the effective stress and the new probabilities obtained is reported in Figure 7.25 as ADMVFO with new perturbations.

Next, the results of radial displacement (x-component) response variable are presented. The mean value of the response variable is presented in the form of contour plots Figure 7.26. The standard deviation of the radial displacement is displayed in the form of contour plots shown in Figure 7.27. Though the magnitudes of the standard deviations are small, the trailing edge of the airfoil shows the largest deviation (Figure 7.27). Sensitivity factors of the radial displacement to the random variables considered are shown in the form of contour plots in Figure 7.28 through Figure 7.36. The sensitivities point out that the radial probabilistic displacement at the trailing edge of

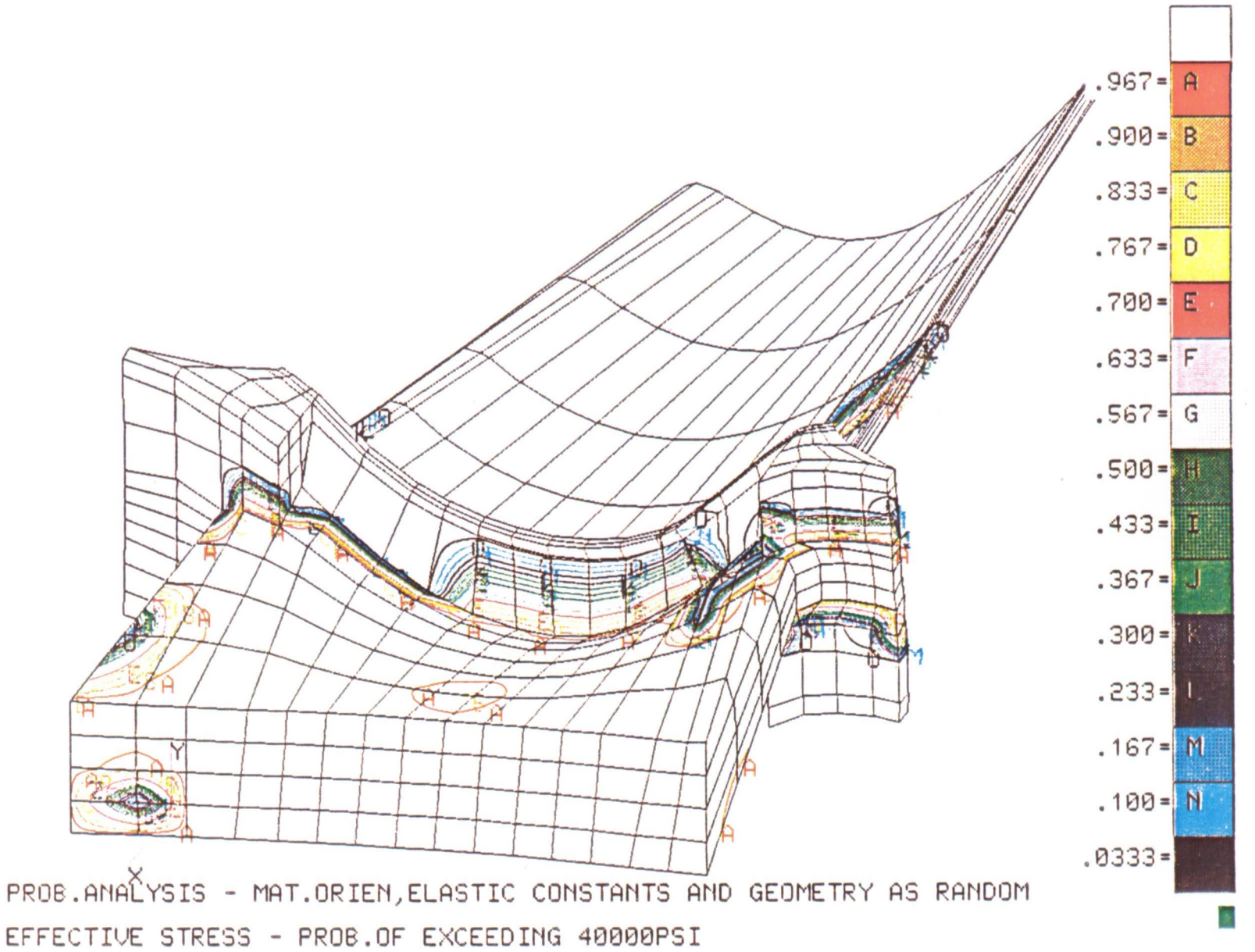


Figure 7.17 Probabilities Of Exceedance Of The Effective Stress To The 40000 PSI Stress Level

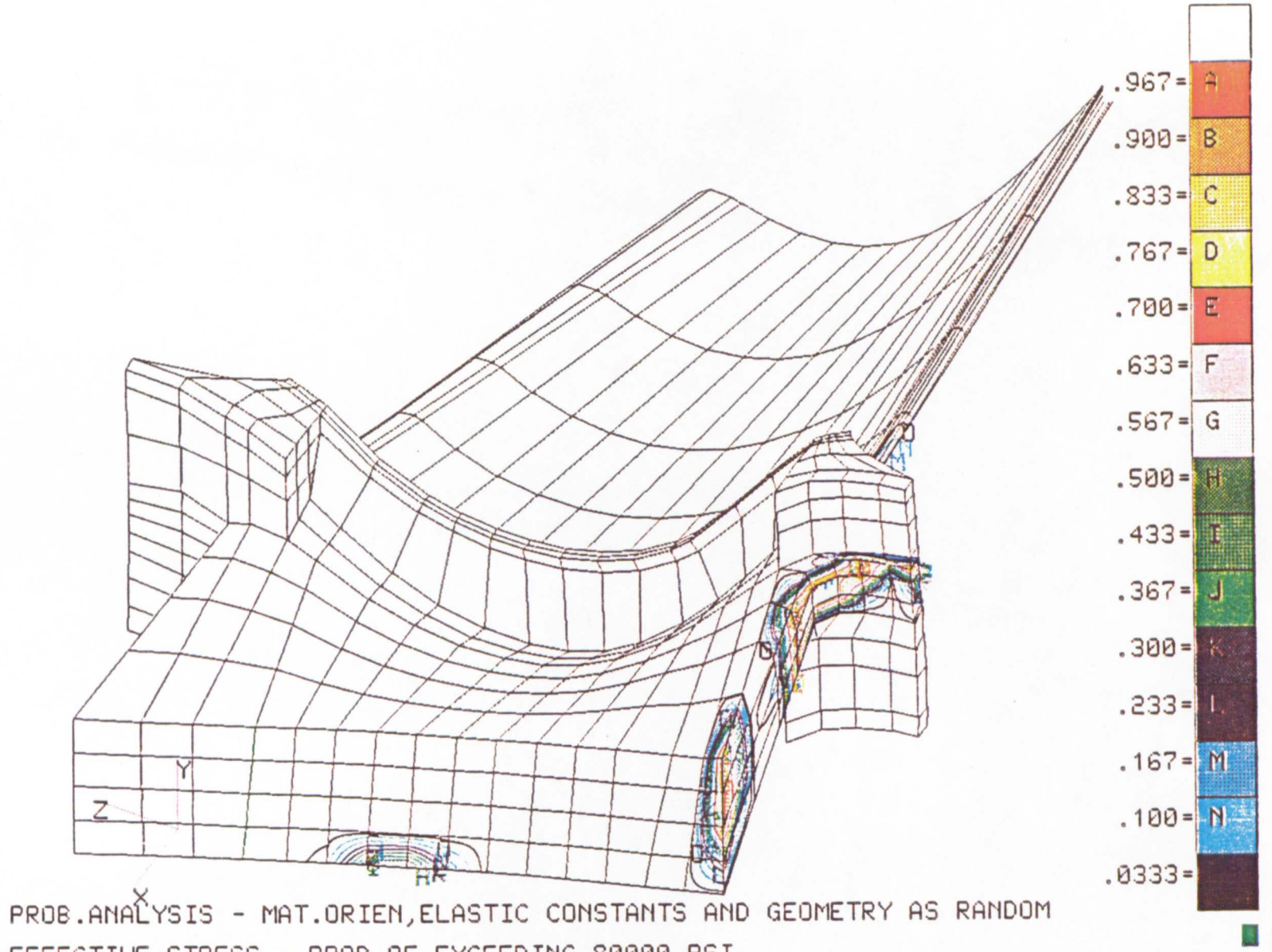


Figure 7.18 Probabilities of Exceedance of the Effective Stress to the 80000 psi Stress Level

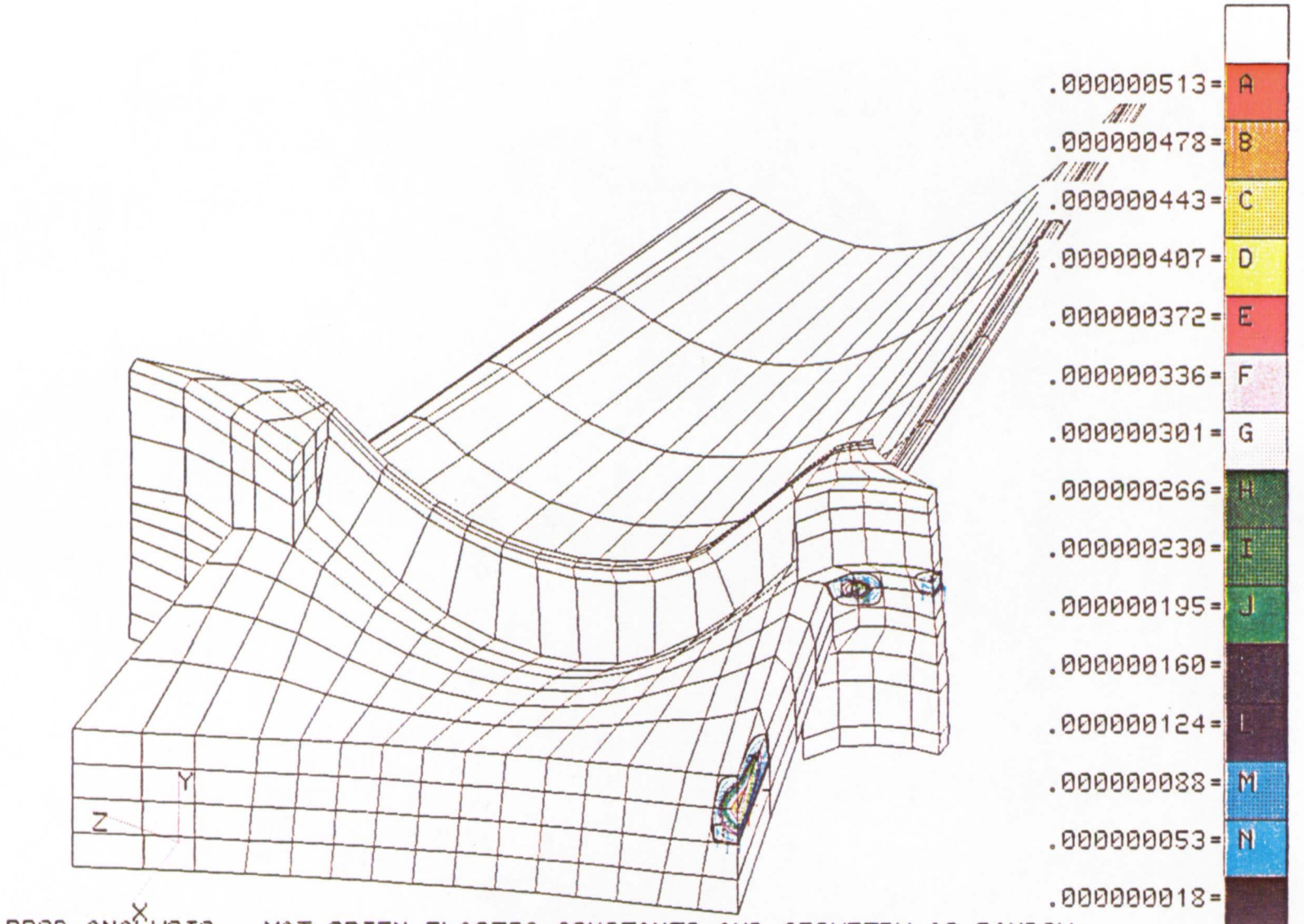


Figure 7.19 Probabilities of Exceedance of the Effective Stress to the 120000 psi Stress Level

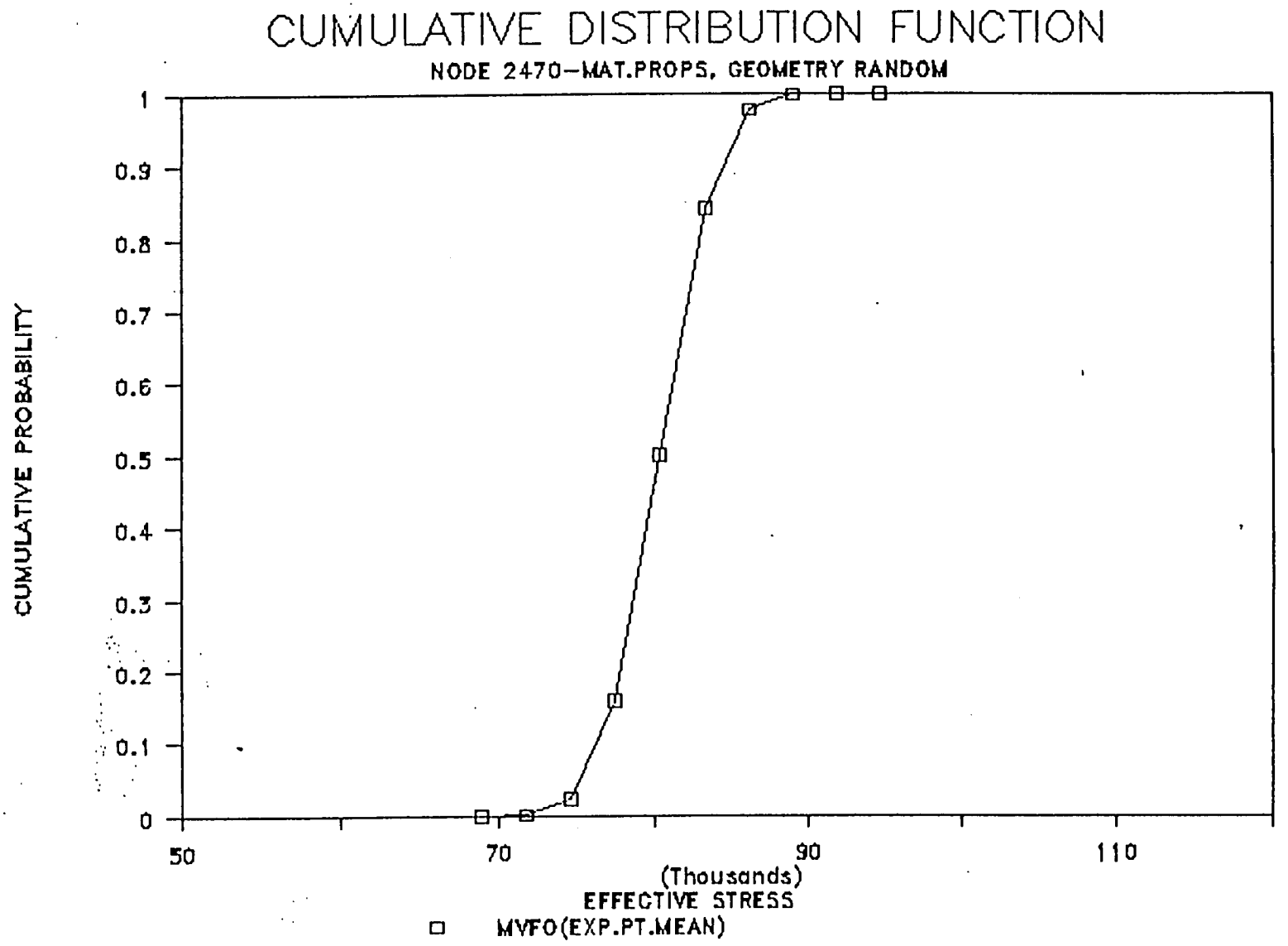


Figure 7.20 Cumulative Probability Function For Effective Stress At Node 2470

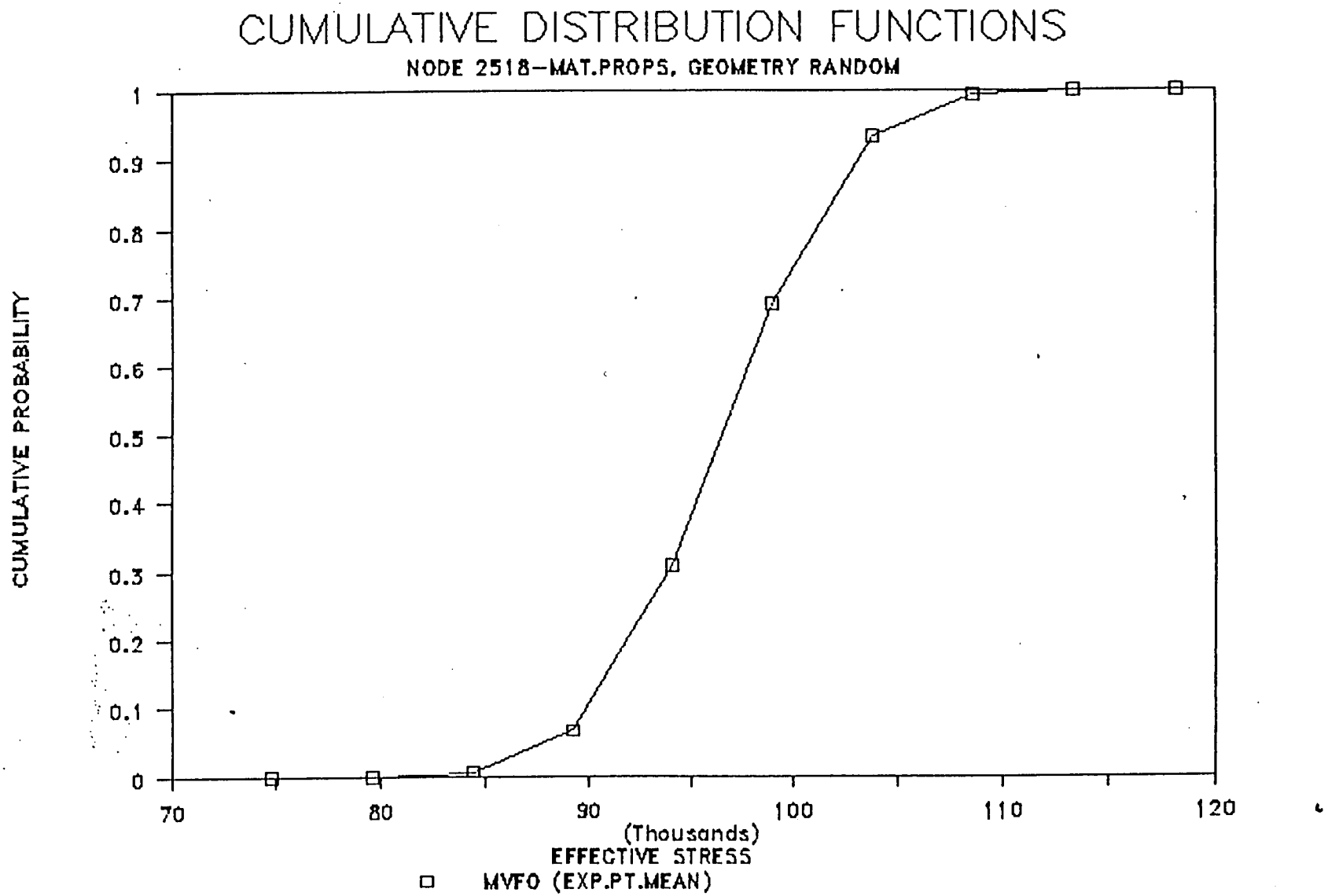


Figure 7.21 Cumulative Probability Function For Effective Stress At Node 2518

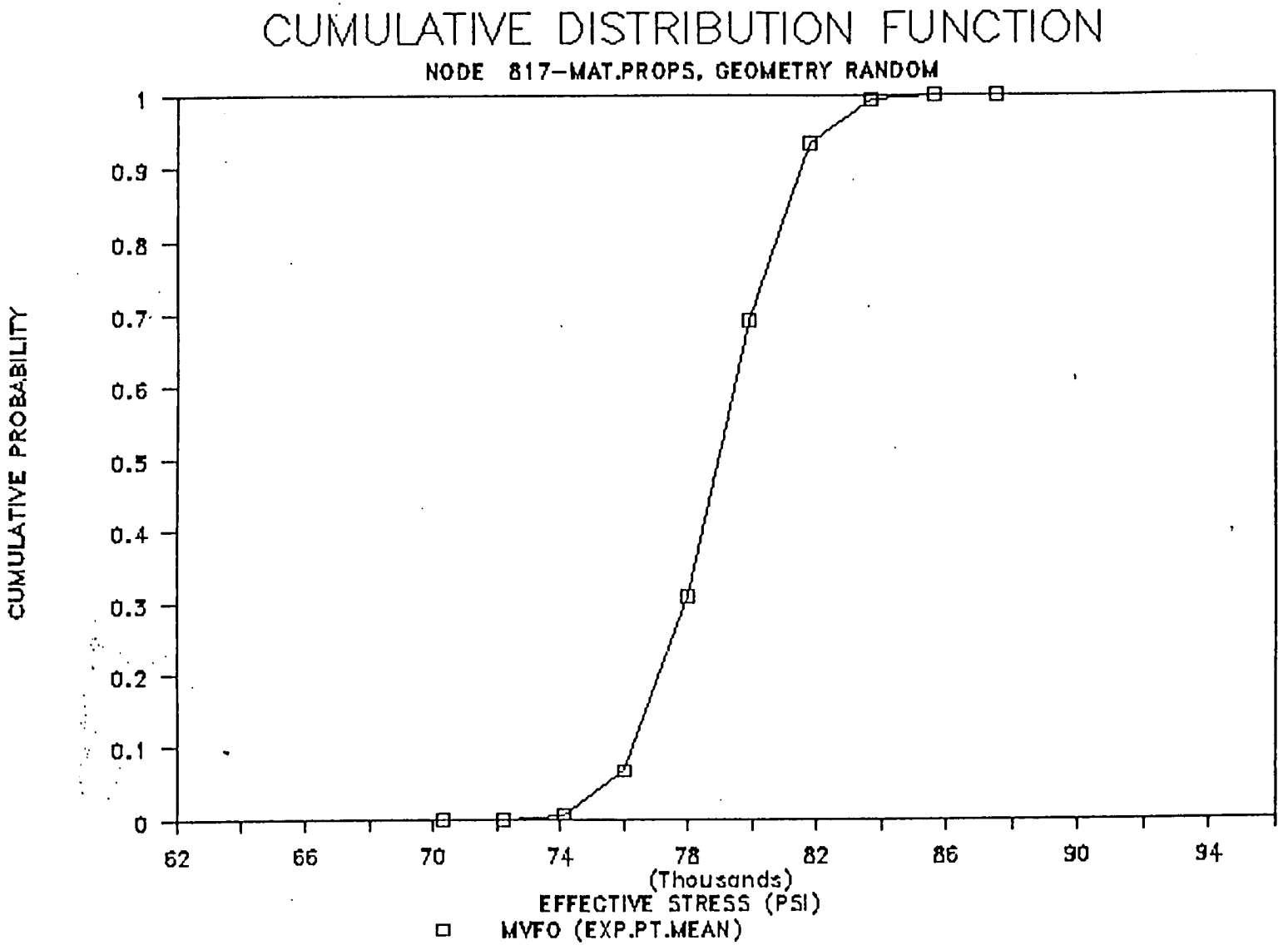


Figure 7.22 Cumulative Probability Function for Effective Stress at Node 817

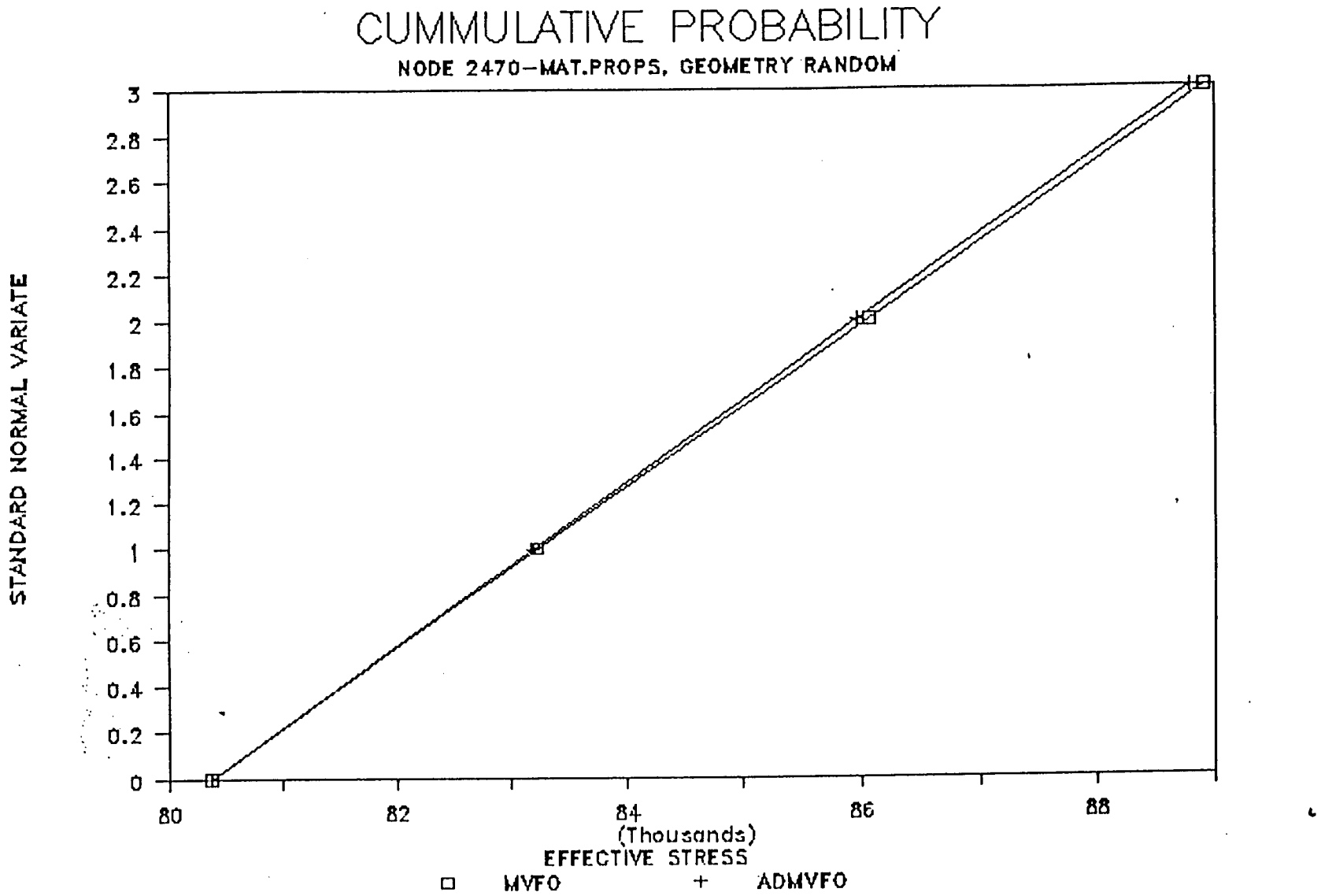


Figure 7.23 Probability Levels Comparison Between MVFO and ADMVFO Method for Node 2470

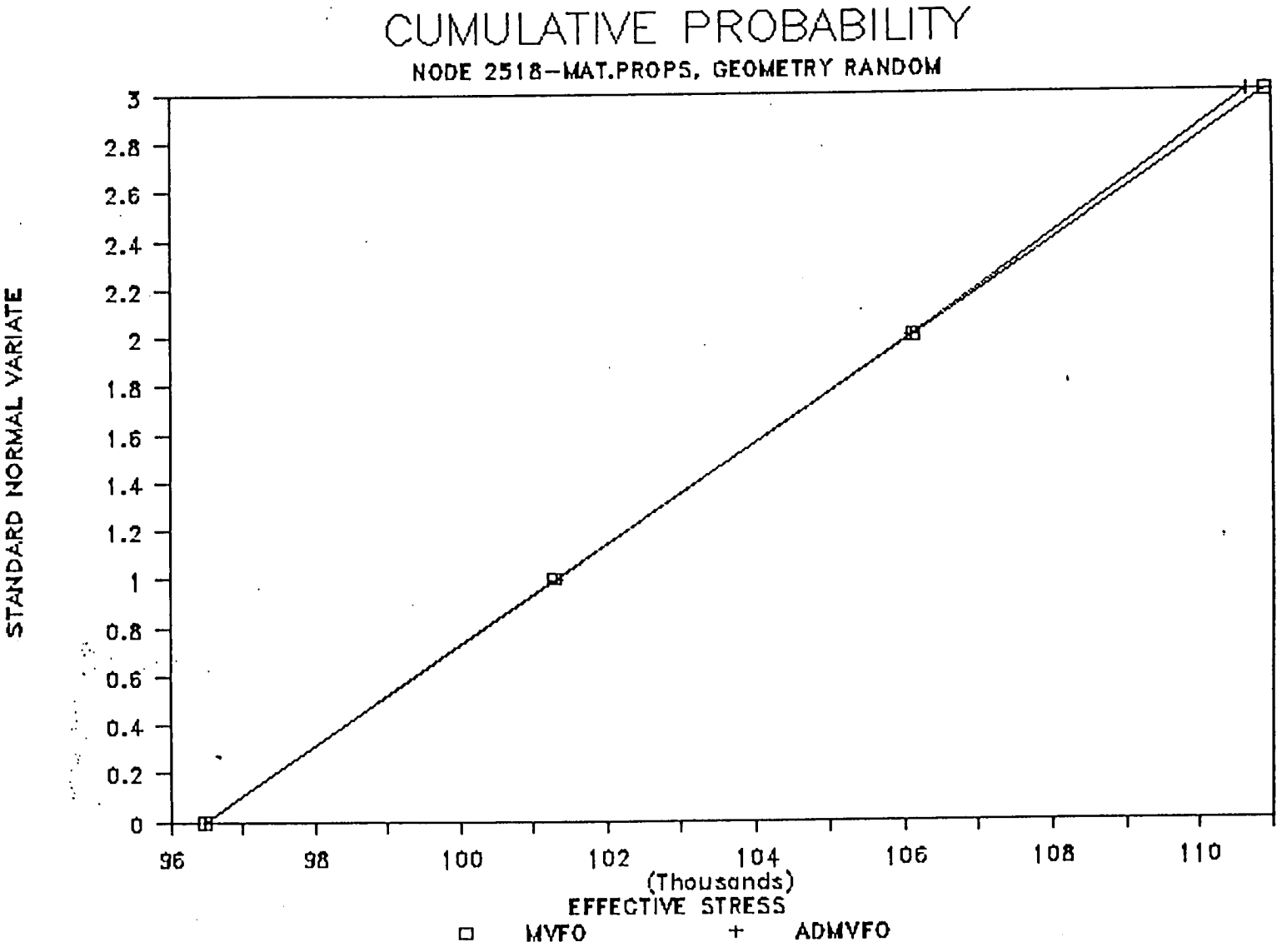


Figure 7.24 Probability Levels Comparison Between MVFO Method and ADMVFO Method for Node 2518

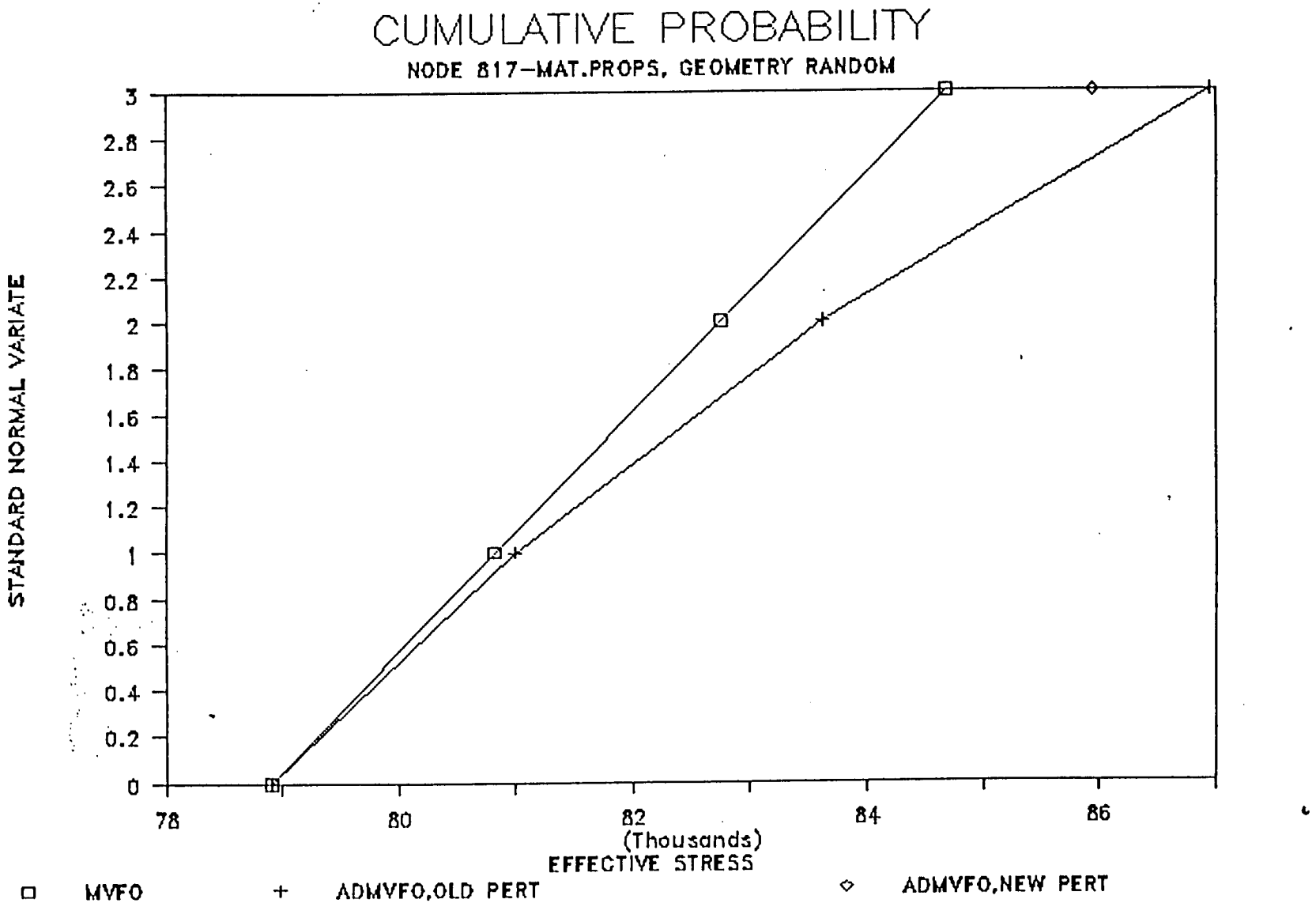


Figure 7.25 Probability Level Comparison Between MVFO Method and Advanced MVFO Method for Node 817

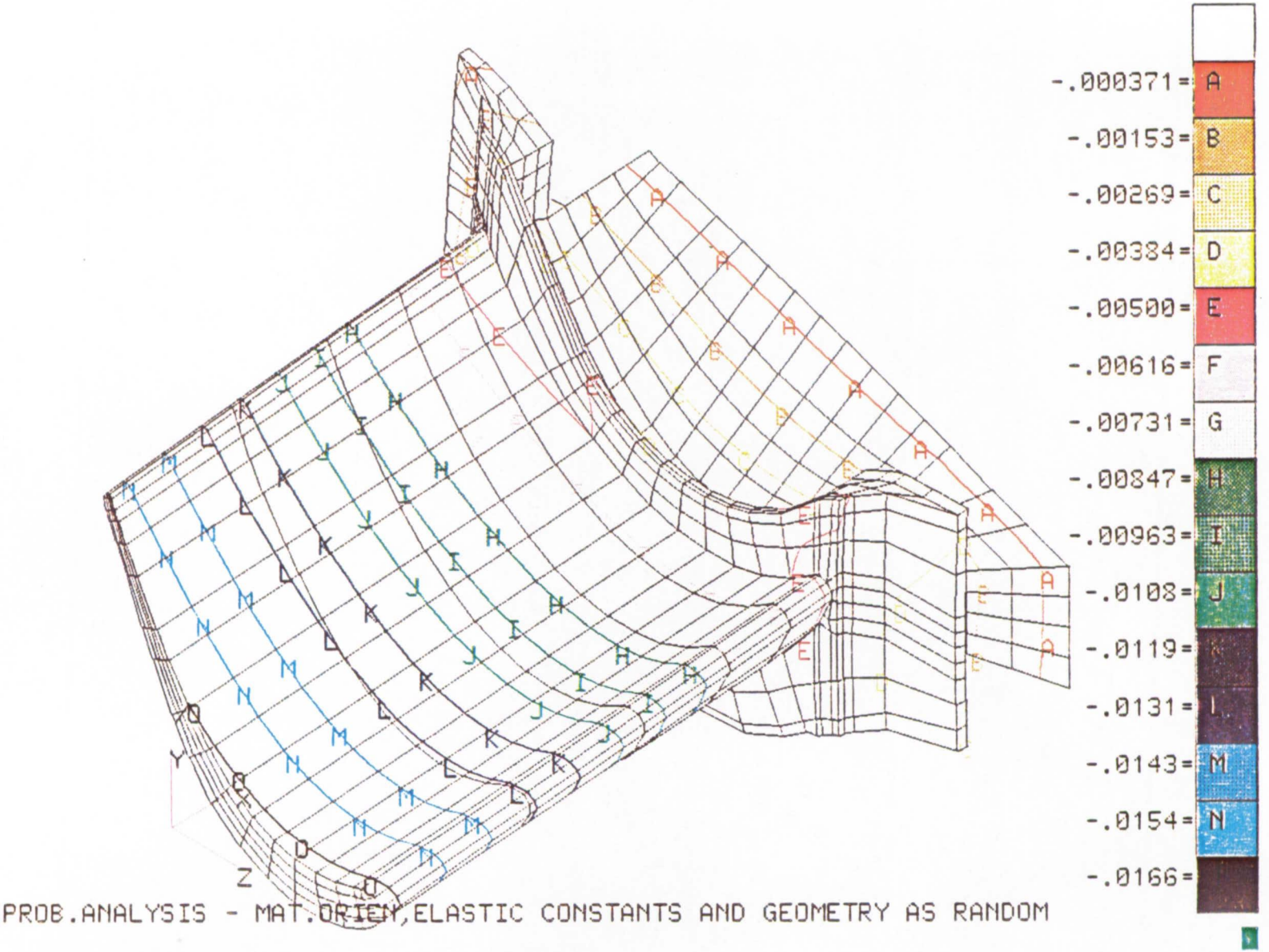


Figure 7.26 Radial Displacement Mean Values

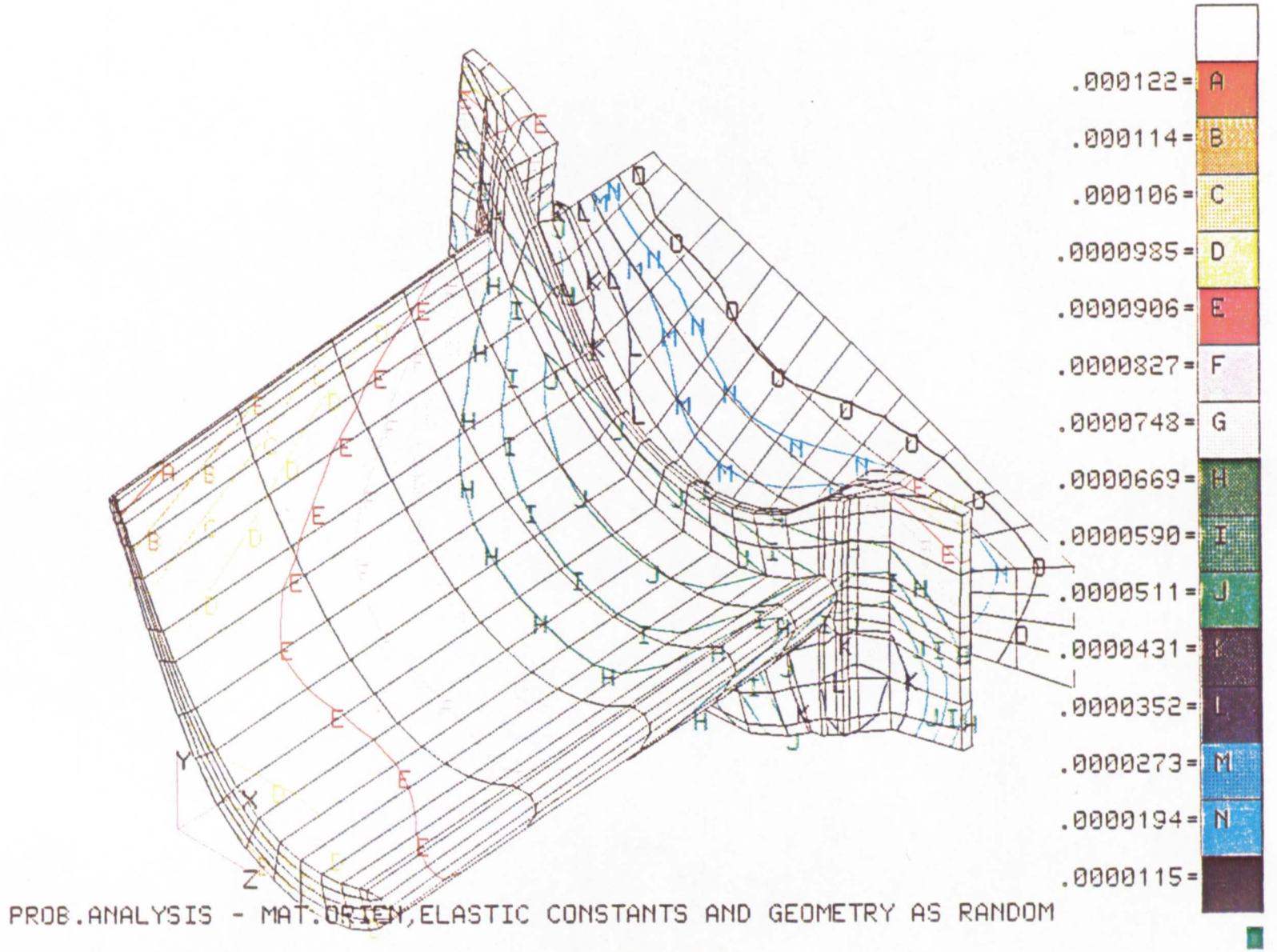


Figure 7.27 Radial Displacement Standard Deviations

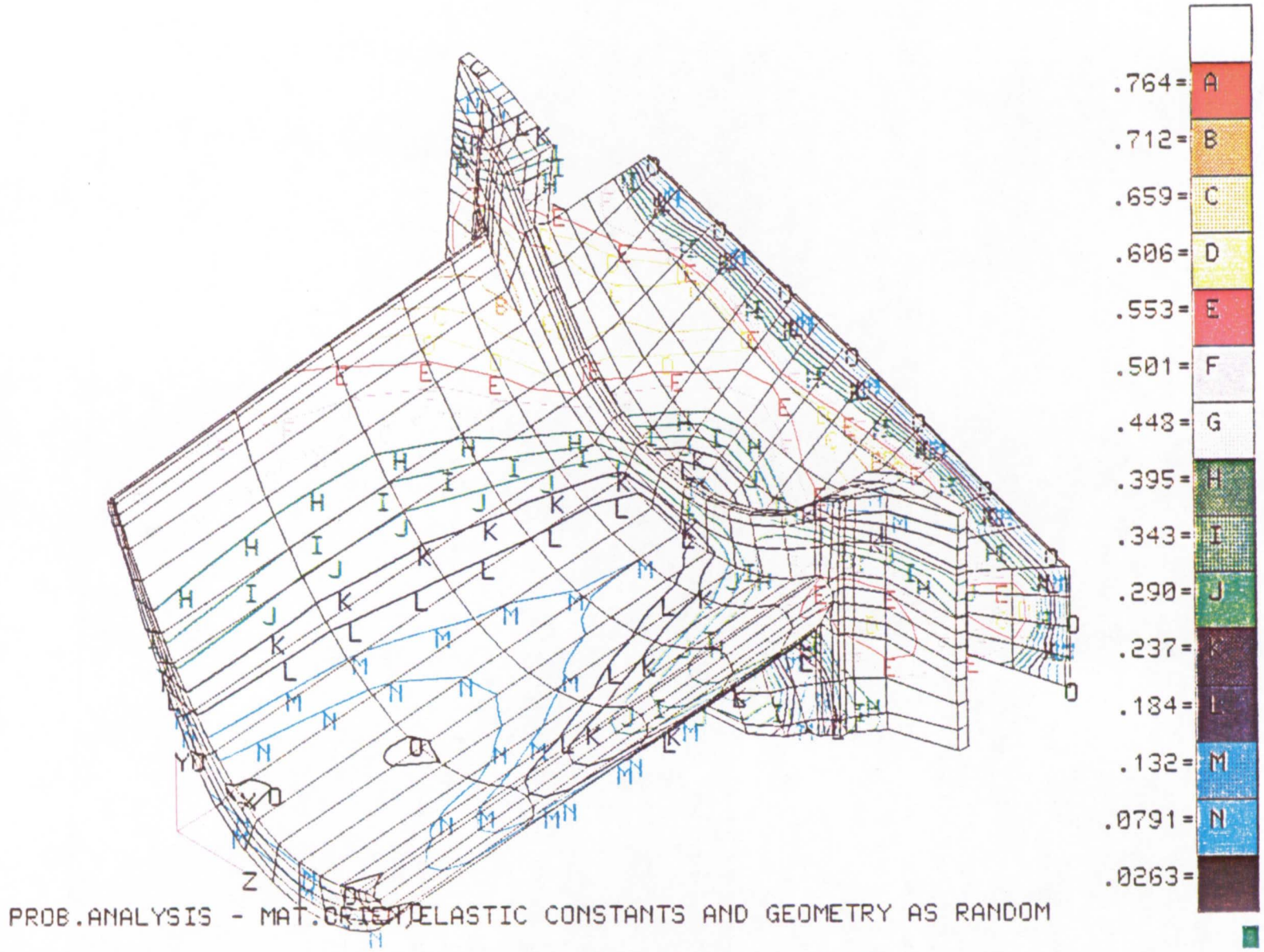


Figure 7.28 Radial Displacement Sensitivity To Material Axis Lean Angle

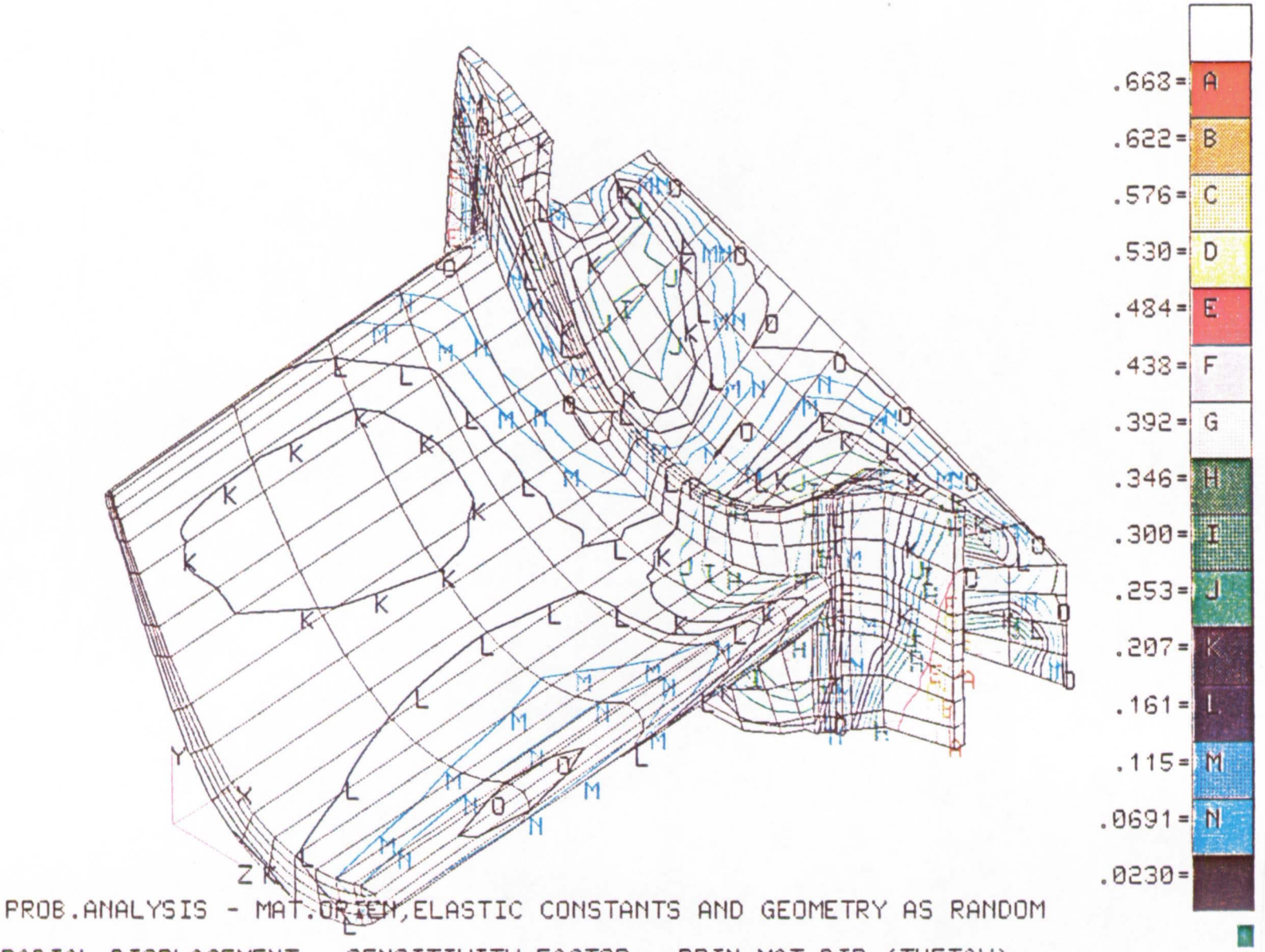


Figure 7.29 Radial Displacement Sensitivity To Material Axis Tilt Angle

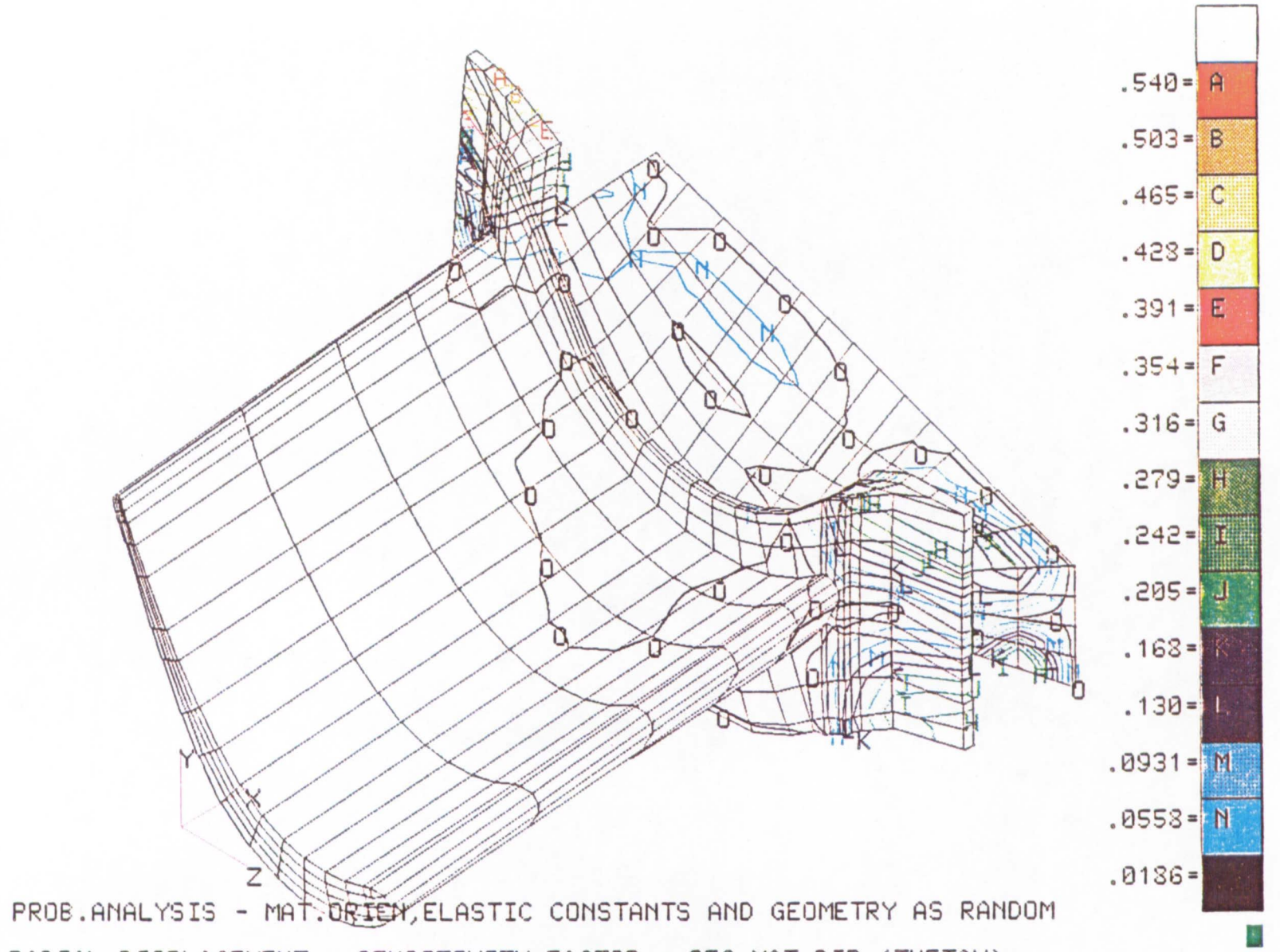


Figure 7.30 Radial Displacement Sensitivity to Material Axis Twist Angle

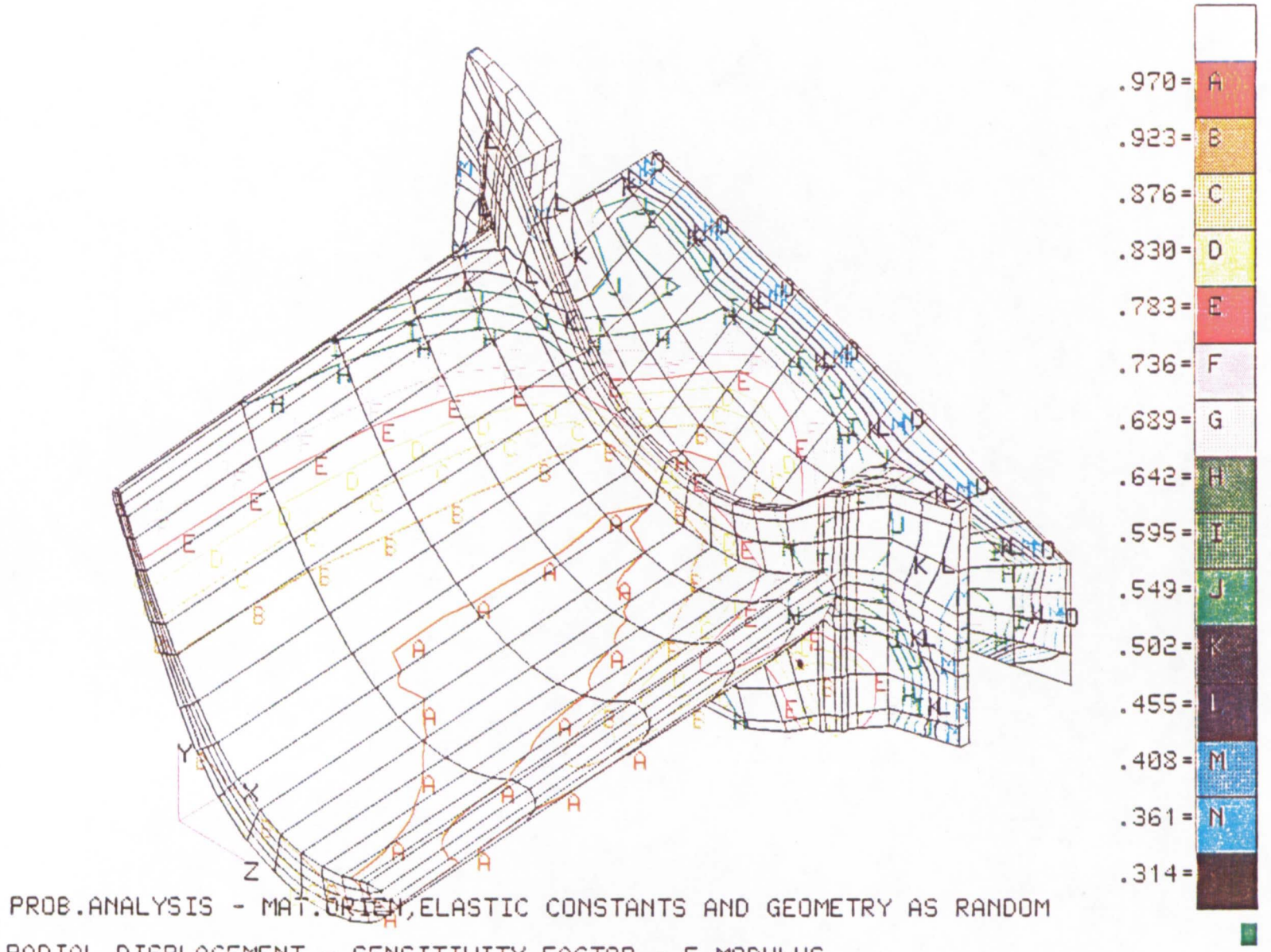


Figure 7.31 Radial Displacement Sensitivity to Young's Modulus

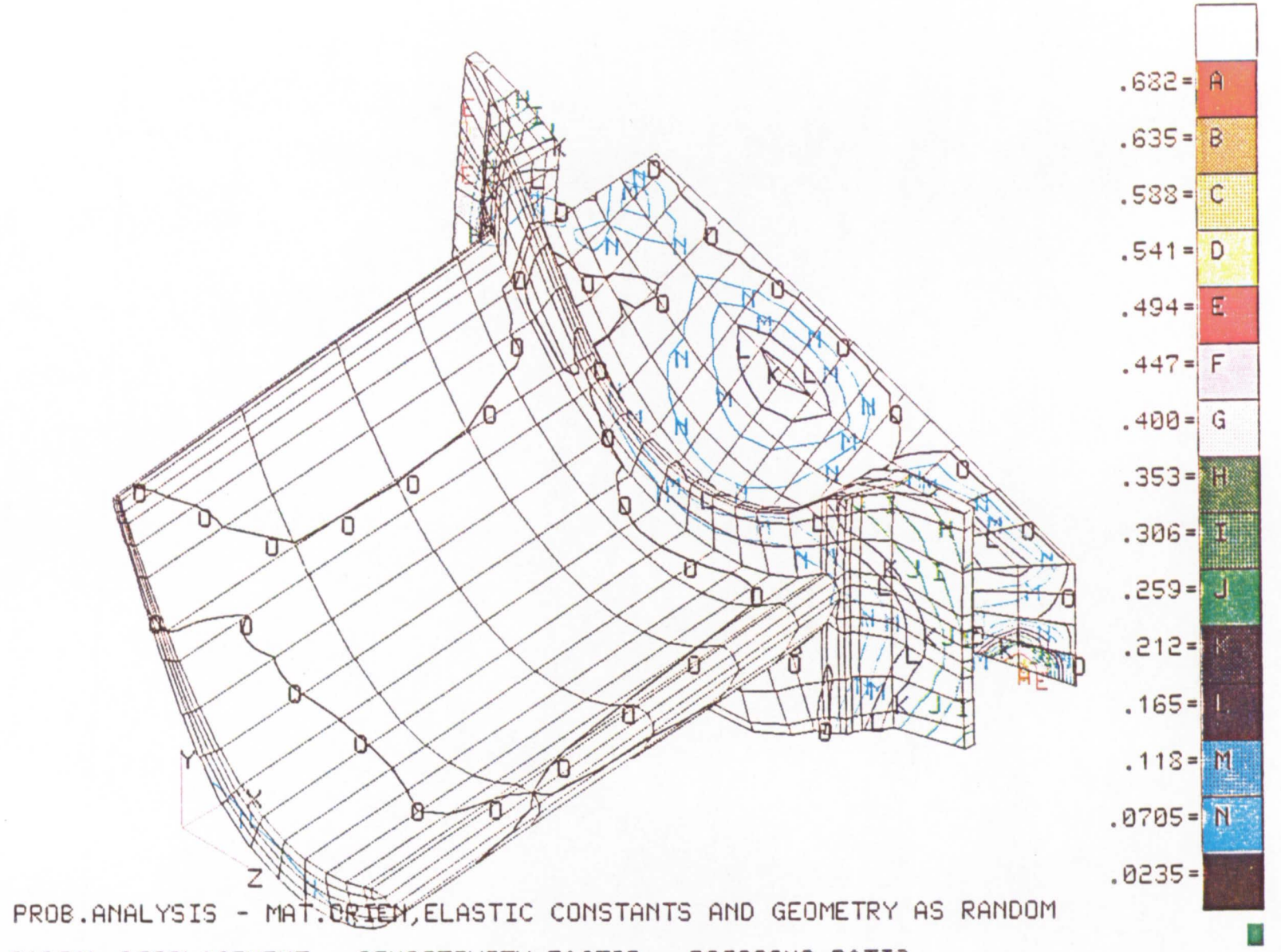


Figure 7.32 Radial Displacement Sensitivity to Poisson's Ratio

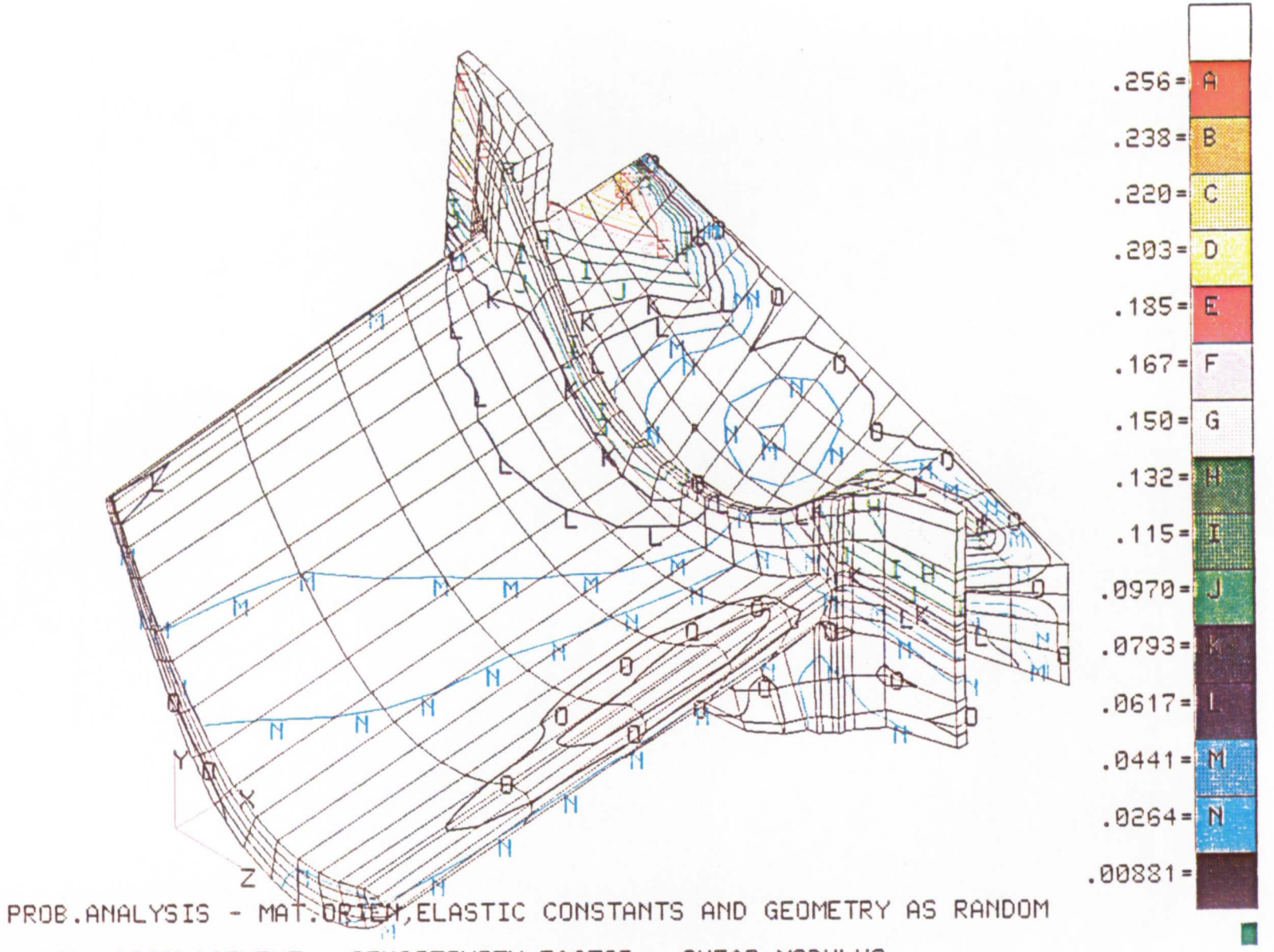


Figure 7.33 Radial Displacement Sensitivity to Shear Modulus

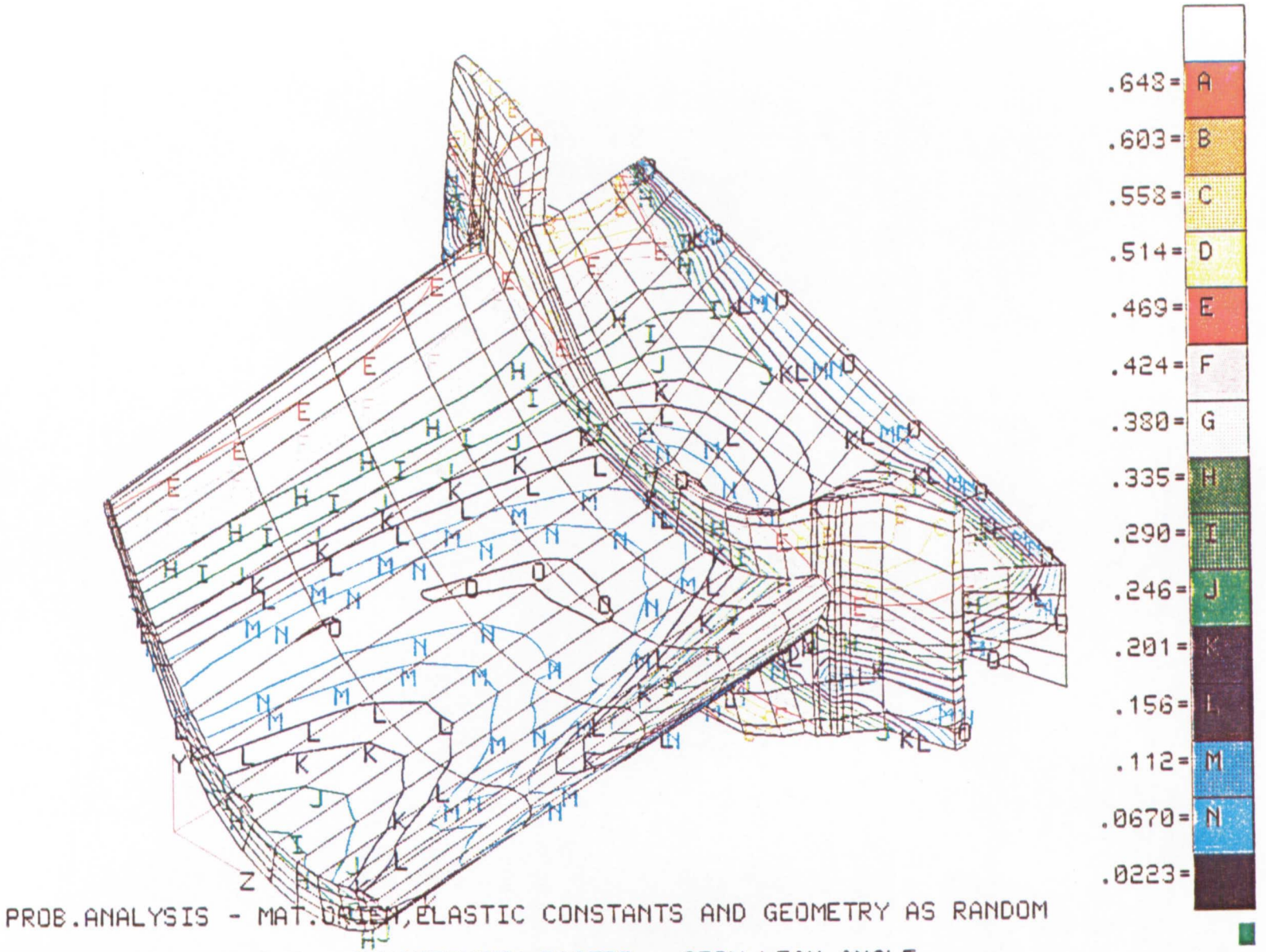


Figure 7.34 Radial Displacement Sensitivity to Geometric Perturbation Lean Angle

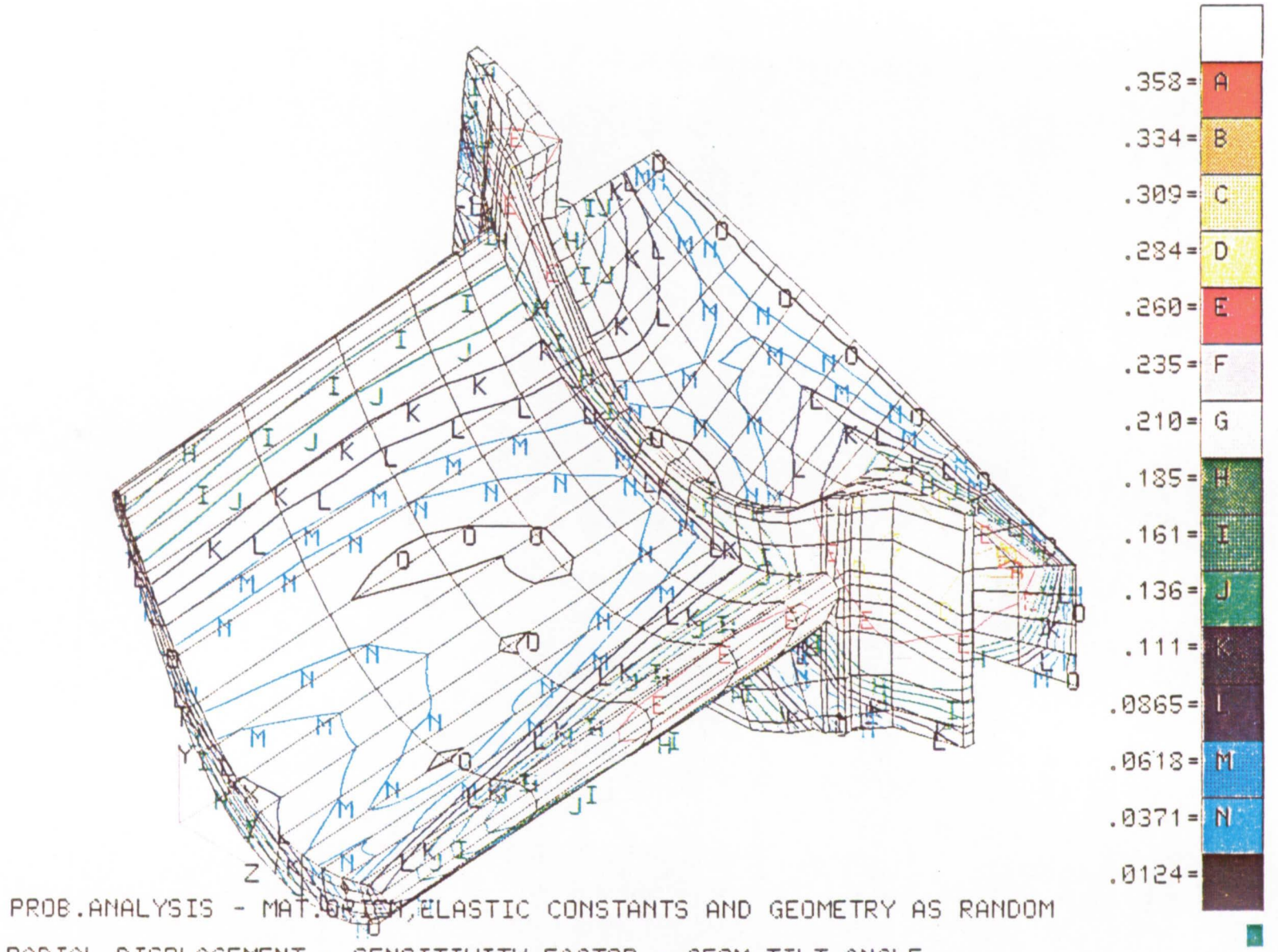


Figure 7.35 Radial Displacement Sensitivity to Geometric Perturbation Tilt Angle

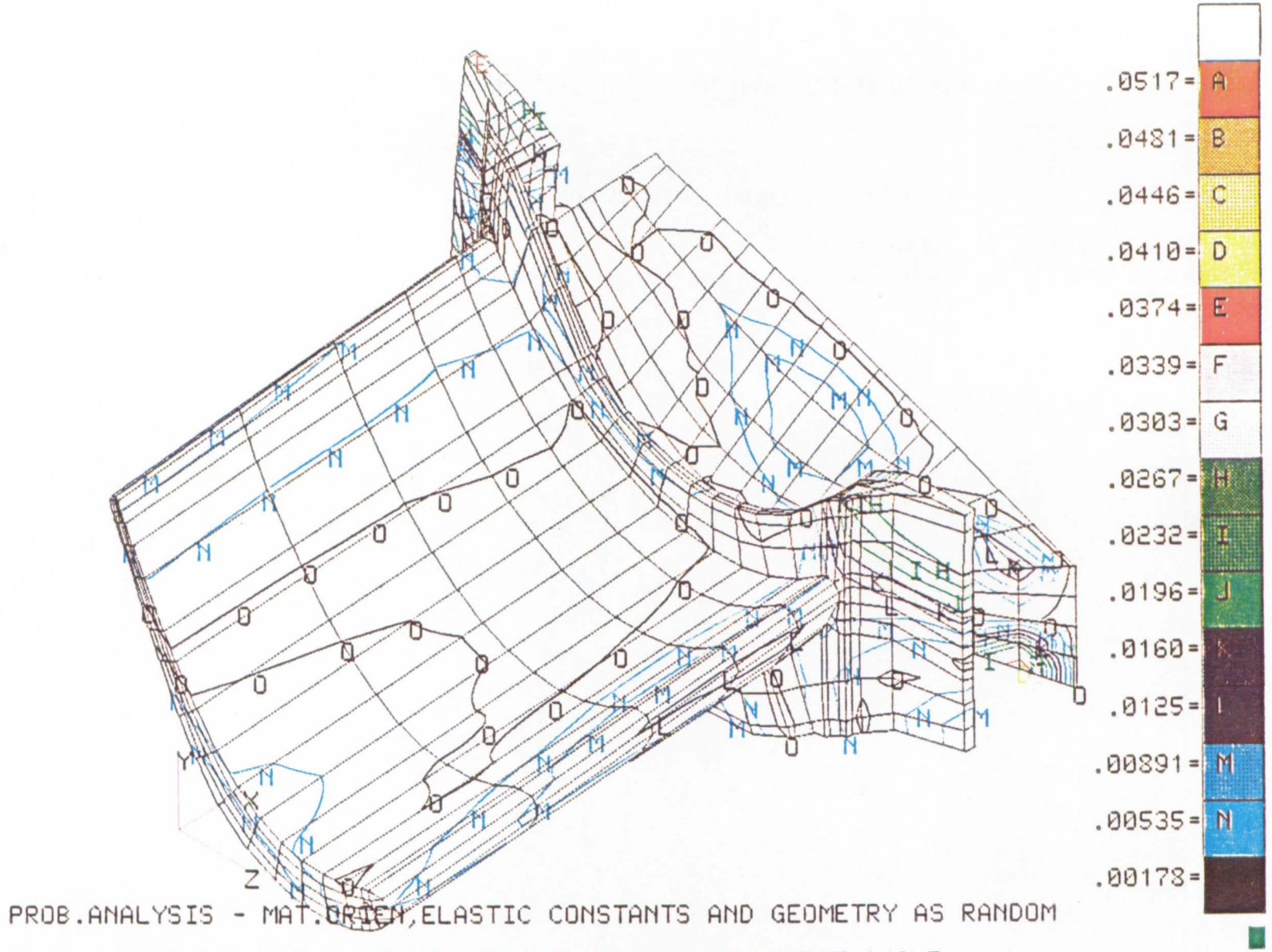


Figure 7.36 Radial Displacement Sensitivity to Geometric Perturbation Twist Angle

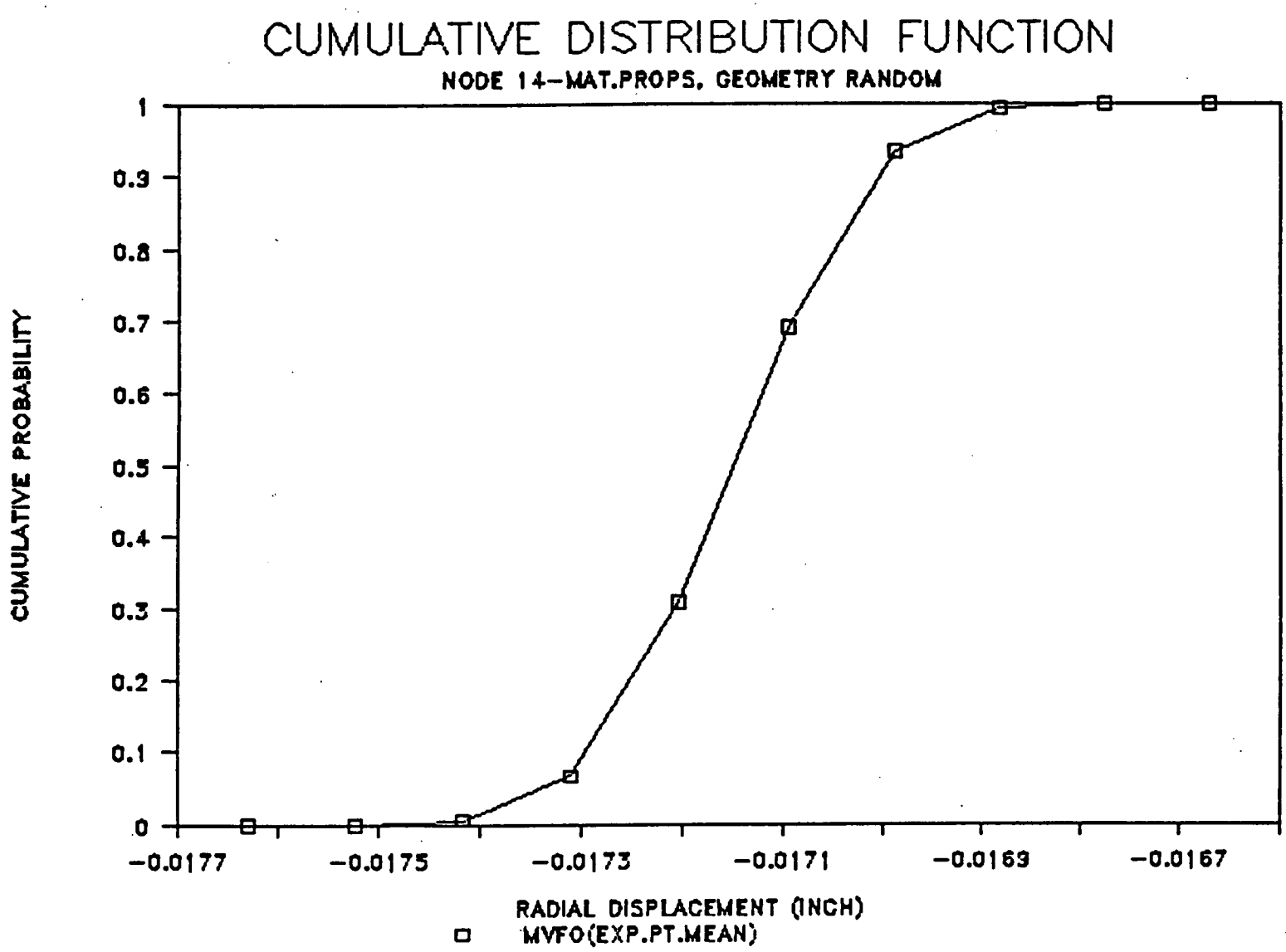


Figure 7.37 Cumulative Probability Function for Radial Displacement at Node 14

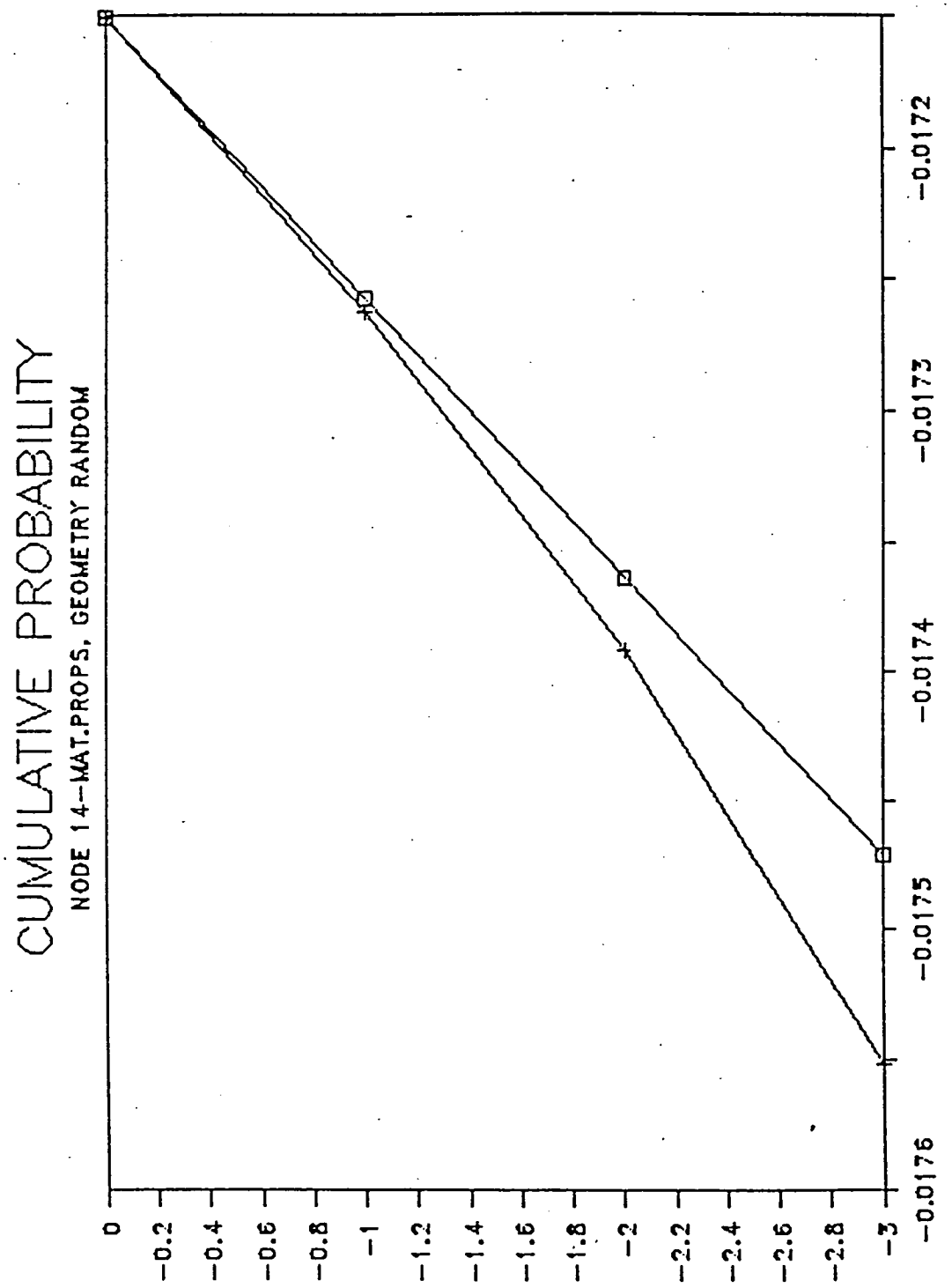


Figure 7.38 Comparison of Probabilities from MVFO and Advanced MVFO Method for Radial Displacements at Node 14

the airfoil is primarily affected by the elastic modulus and the primary material axes orientation random variables.

A node having the maximum radial tip displacement (node 14) was chosen for further processing. The cumulative distribution function for the tip displacement based on the MVFO method is shown in Figure 7.37. The cumulative probabilities are plotted in a different form in Figure 7.38. The response variable was further processed using advanced fast probability method in which the design points were successively moved to -1 sigma, -2 sigma, and -3 sigma values. NESSUS/FPI was used again to calculate the new probabilities but with old perturbation data. The results are reported in Figure 7.38 as the advanced mean-value, first-order method. Further comparison between the results for the blade verification analysis and validation cases will be made in the future.

7.4 NESSUS/FEM and NESSUS/FPI Computational Experience

The NESSUS/FEM code has been executed in a variety of computers during the verification, validation and check out phases. As computational cost is of much interest to the end user some of the computational statistics are reported in Table 7.3. The details of the blade finite element model used in the verification efforts are shown in Figure 7.2.

Table 7.3

Blade Verification Study Run Time Statistics

CRAY XMP1-4 COS	IBM 3090 MVS	CDC 990 NOS-VE	CDC 860 NOS-VE	ALLIANT FX-8 1-CU UNIX
A	A/B	A/B	B	A
185	280/364	370/460	2890	2445

A = Vectorized B = Non-Vectorized

NESSUS/FPI 2519 NODES - MVFO METHOD - 330 SECS (CDC990 NOS-VE)

It has been observed that, for this verification problem, the computational solution times for element formulation, equation solution and stress recovery phases in NESSUS are comparable to the commercial codes such as ANSYS. However, the band width optimizer module is inefficient and takes

an inordinate percentage of the total solution time. For the case of blade verification, nearly 40% of the computational time for the deterministic analysis was spent on bandwidth optimization modules. Without specifically tuning the FORTRAN code for any specific compiler on a particular machine, a 20% to 25% increase in computational speed is obtained by vectorizing. For the blade verification problem, each perturbation solution took about two to three iterations to converge. Computation time for each converged perturbation solution was approximately 50% of the corresponding deterministic solution. Improvements in this ratio can be expected when the node optimizer module is improved.

While all the verification studies conducted so far used the strict isoparametric Type 7 elements, the newly developed element Type 154 was also exercised on the blade verification model for selected cases. The results indicated for the Type 154 element the stiffness formulation times were approximately 2 times that of element Type 7. The stress recovery and perturbation iteration phases were approximatly 2.5 times more time-consuming compared to element Type 7. While element Type 154 provided improved results, improvements in the computational speed for Type 154 modules is recommended. As part of the preliminary verification process, the mixed iteration technique was also exercised. It was found that a combination of mixed iteration and multiple perturbation solutions for the size of the blade verification problem was considered excessive CPU time-consuming and, therefore, was not used. The standard displacement solution was used throughout the verification studies.

Minor modifications to the NESSUS/FPI code allowed the processing of all nodes in the verification model for a given response variable for the MVFO method. The cost of the solution which allows to process the entire model using MVFO method is equivalent to a single deterministic FEM solution. Based on the current experience for the size of models considered under the blade verification study, it is unrealistic to expect to process the entire model using the ADMVFO method for different probability levels. This is because of the continuous shift of the design points to obtain new deterministic solutions and the new gradients around the design point for each probability level and node point. However, new techniques and strategies using iterative solutions to obtain values corresponding to new design points might be worth pursuing to reduce the computational cost. It is feasible now to process selected critical nodes using ADMVFO method for many probability levels.

7.5 Fiscal Year 1988 Effort

Blade verification studies will continue during FY 88. The load variables of pressure, temperature and speed will be treated as random. The method of treatment of these load variables for the blade has been obtained from the composite load spectra contract. The perturbation results from the load variables will be added to the existing database to reanalyze the response variable presented in this report.

Initial verification studies conducted on a simplistic model pointed out the shortcoming and errors in deterministic frequency extraction as well as in the method of frequency extraction for the perturbed structure. The new NESSUS 2.5/FEM release which has enhancements and bugs removed for this phase of analysis will be used for probabilistic analysis of frequencies in turbine blades. The additional random variable to be considered for this phase would include mass density. A method for considering support stiffness variations will be studied.

The verification efforts for the duct component will begin. The primary analysis will be random vibration analysis with vibration levels, the structure properties, and geometry considered as random. Initial verification efforts will define the enhancements if any needed in NESSUS/FEM followed by the full verification analysis.

8.0 A STRESS-BASED HYBRID FINITE ELEMENT METHOD FOR ELASTO-PLASTIC ANALYSIS USING THE ENDOCHRONIC THEORY

8.1 Introduction

In this chapter, a three dimensional 16-node stress-based hybrid finite element for shell structural analysis will be formulated, using the endochronic elasto-plastic constitutive theory. The iterative scheme for the solution of the nonlinear system of equations that results will be presented, with the mid-point radial return algorithm being used to improve the accuracy of the integrations.

The motivation for the stress-based element is predicated on the observation that the assumed-stress hybrid model has been demonstrated to give more accurate displacements and stress solutions than the conventional displacement model [1]. In general, for shell analysis the degenerated shell element is often used. However, in such an element the nature of stresses, strains, and displacements is limited to a linear variation through the thickness, which may not be the situation in complicated problems of loading. On the other hand, conventional displacement-based three dimensional solid element can present well, all of the physical fields in the in-plane directions as well as in the through thickness direction, but can not tolerate higher aspect ratios (i.e. the case when the thickness, compared with other two dimensions, is too small). In a recent study [2], it is shown that the stress-based hybrid element, in addition to providing better stresses, can also sustain much more severe distortions than the displacement element. Furthermore, due to the more accurate stress solution, the use of the hybrid stress model for nonlinear problems, where the nonlinearity arises from the coupling of material behavior to the stress field, should result in a faster rate of convergence.

8.2 Stress-Based Hybrid Finite Element Formulation

8.2.1 Assumptions for the Hybrid Formulation

Here, it is assumed that the loads and/or displacements are applied incrementally, and one must satisfy the following equations within the volume of the element:

Linear Momentum Balance;

$$\Delta\sigma_{ij,j} + \Delta f_i = 0 \qquad \text{in } V_m \tag{8.1}$$

Angular Momentum Balance;

$$\Delta\sigma_{ij} = \Delta\sigma_{ji} \qquad \text{in } V_m \tag{8.2}$$

Strain - Displacement Relation;

$$\Delta\varepsilon_{ij} = \frac{1}{2}(\Delta u_{ij} + \Delta u_{j,i}) \qquad \text{in } V_m \tag{8.3}$$

Constitutive Relation;

$$\Delta\varepsilon_{ij} = S_{ijk\ell}\,\Delta\sigma_{k\ell} \tag{8.4}$$

Traction Boundary Condition;

$$\Delta\sigma_{ij}n_j = \Delta\bar{T}_i \qquad \text{at } S_\sigma \tag{8.5}$$

Displacement Boundary condition;

$$\Delta u_i = \Delta\bar{u}_i \qquad \text{at } S_u \tag{8.6}$$

Inter-element Boundary Conditions;

traction reciprocity

$$\Delta\sigma_{ij}n_i^+ + \Delta\sigma_{ij}n_j^- = 0 \qquad \text{at } S_\rho \tag{8.7}$$

displacement

$$u_i^+ = u_i^- \qquad \text{at } S_\rho \tag{8.8}$$

In the above, S_ρ is defined to be on the boundary of the interface of two elements with the total boundary of an element defined as

$$S = S_\rho \cup S_\sigma \cup S_u \tag{8.9}$$

Relaxing the requirement that the stress field within the element satisfy the equilibrium equation a priori, the stress field will be selected to satisfy the angular momentum balance, $\Delta\sigma_{ij} = \Delta\sigma_{ji}$, only. Likewise, assume that the change in strain can be related to the change in stress through:

$$\Delta\varepsilon_{ij} = S_{ijk\ell}\,\Delta\sigma_{k\ell} \tag{8.10}$$

Note that in what follows $S_{ijk\ell}$ was assumed to be composed of an elastic part and a plastic part with

$$\Delta\varepsilon_{ij} = \Delta\varepsilon_{ij}^{e} + \Delta\varepsilon_{ij}^{p} \tag{8.11}$$

$$\Delta\varepsilon_{ij} = S_{ijk\ell}^{e}\,\Delta\sigma_{k\ell} + S_{ijk\ell}^{p}\,\Delta\sigma_{k\ell} \tag{8.12}$$

where

$$S_{ijk\ell}^{e} = \frac{\delta_{ik}\delta_{j\ell}}{2\mu} + \frac{\lambda}{(3\lambda+2\mu)2\mu}\,\delta_{ij}\,\delta_{k\ell} \tag{8.13}$$

$$S_{ijk\ell}^{p} = \frac{N_{ij}N_{k\ell}}{(2\mu)(C-1)} \tag{8.14}$$

8.2.2 Weak Form

Based on the a priori conditions and the enforcement of equilibrium condition, compatibility condition, traction boundary condition and traction reciprocity, the weak form of this stress-based hybrid formulation can expressed as:

$$\sum_m \int_{V_m} \left\{\left(S_{ijk\ell}\Delta\sigma_{k\ell} - \frac{1}{2}\left(\Delta u_{i,j} + \Delta u_{j,i}\right)\right\} \delta\sigma_{ij}\, dV$$

$$+ \int_{S_\sigma} \left(\Delta\sigma_{ij} n_j - \Delta T_i\right) \delta u_i\, dS + \int_{S_\sigma} \Delta\sigma_{ij} n_j \delta u_i\, dS \tag{8.15}$$

$$+ \int_{V_m} \left(\Delta\sigma_{ij,j} + \Delta f_i\right) \delta u_i\, dV = 0$$

Which after applying Green's theorem reduces to:

$$\sum_m \int_{V_m} S_{ijk\ell}\Delta\sigma_{k\ell}\Delta\sigma_{ij}\, dV + \int_{V_m} \Delta u_{i,j}\, \delta\sigma_{ij}\, dV + \int_{V_m} \delta u_{i,j}\Delta\sigma_{ij}\, dV \qquad (8.16)$$

$$+ \int_{S_\sigma} \Delta T_i \delta u_i\, dS + \int_{V_m} \Delta f_i\, \delta u_i\, dV = 0$$

8.2.3 Discrete Weak Form

The stresses within an element were represented as a summation of polynomial stress modes, A_{ijm}, with undetermined parameters $\Delta\beta_m$;

$$\Delta\sigma_{ij} = A_{ijm}\,\Delta\beta_m \qquad (8.17)$$

Refer to Appendix B for the exact form of the polynomials used to formulate the hybrid element. To enforce compatibility in a weak form, one may use a test function of the same class as the function for stress. Define $\delta\sigma_{ij}$ as the test function in terms of the same polynomial stress modes, A_{ijm}, with the arbitrary parameters.

$$\delta\sigma_{ij} = A_{ijm}\delta\beta_m \qquad (8.18)$$

The displacements were interpolated from the nodal values, Δq_k, and the standard isoparametric shape functions, ψ_{ik}, as:

$$\Delta u_i = \psi_{ik}\Delta q_k \qquad (8.19)$$

The trial functions for the displacements and the stresses were expressed in terms of the parameters $\Delta\beta_m$ and Δq_k. Define

$$\delta u_i = \psi_{ik}\delta q_k \qquad (8.20)$$

as the test function in terms of the shape functions used in the interpolation of the displacements. The parameters, δq_k, will be arbitrary except on the portion of the boundary where the displacements are prescribed, in which case they will be zero.

Substitution of the discrete form for the test and trial functions (Eq. 8.17-8.20) expressed the combined weak form in terms of $\Delta\beta_m$ and Δq_k to give:

$$\sum_m \int_{V_m} \Delta\beta_m A_{ijm} S_{ijk\ell} A_{k\ell n} \delta\beta_n dV + \int_S \Delta q_k \psi_{ik} n_j A_{ijm} \delta\beta_m dS + \tag{8.21}$$

$$\int_S \Delta\beta_m A_{ijm} n_j \psi_{ik} \delta q_k dS + \int_{S_{\sigma_m}} - \Delta T_i \psi_{ik} \delta q_k dS = 0$$

Defining the matrix

$$\underline{\underline{H}} = \int_{V_m} A_{ijm} S_{ijk\ell} A_{k\ell n} dV \tag{8.22}$$

and the matrix

$$\underline{\underline{G}} = \int_S \psi_{ik} n_j A_{ijm} dS \tag{8.23}$$

One may express the combined weak form in matrix form as

$$\sum_m \Delta\underline{\beta}^T \underline{\underline{H}} \delta\underline{\beta} + \Delta\underline{q}^T \underline{\underline{G}} \delta\underline{\beta} + \delta\underline{q}^T \underline{\underline{G}} \Delta\underline{\beta} = \Delta\underline{T}^T \delta\underline{q} + \Delta\underline{F}^T \delta\underline{q} \tag{8.24a}$$

where

$$\Delta\underline{T} = \int \Delta T_i \psi_{ik} dS \; ; \; \Delta\underline{F} = \int \Delta f_i \psi_{ik} dV \tag{8.24b}$$

The global stiffness matrix may be assembled with $\Delta\underline{\beta}$ and $\Delta\underline{q}$ retained as unknowns. The number of unknowns at the global level may be reduced by eliminating the stress parameters which are assumed only within the domain of the element, with no coupling between elements. For arbitrary $\delta\underline{\beta}$ one must satisfy

$$-\underline{\underline{H}} \Delta\underline{\beta} + \underline{\underline{G}} \Delta\underline{q} = \underline{0} \tag{8.25}$$

Thus, $\Delta\underline{\beta}$ may be expressed in terms of the displacement of the element as

$$\Delta \underline{\beta} = \underline{\underline{H}}^{-1} \underline{\underline{G}} \, \Delta \underline{q} \tag{8.26}$$

For arbitrary $\delta \underline{q}$, then one must have at the global level

$$\sum_{m} \underline{\underline{G}}^{T} \Delta \underline{\beta} = \Delta \underline{T} + \Delta \underline{F} \tag{8.27a}$$

or

$$\sum_{m} \underline{\underline{G}}^{T} \underline{\underline{H}}^{-1} \underline{\underline{G}} \, \Delta \underline{q} = \Delta \underline{T} + \Delta \underline{F} \tag{8.27b}$$

8.2.4 Constitutive Modeling

While, until a few years back, simple constitutive relations such as isotropic hardening or linear kinematic hardening plasticity were the mainstay of computer programs, currently there is a widespread interest in the constitutive modeling of experimentally observed behavior of materials involving plastic and creep deformations under monotonic and cyclic loading.

The general theory of internal variables has played a key role in the development of more and more realistic constitutive models to characterize inelastic material behavior. Typical internal variables that are widely employed include: i) the so-called 'back-stress' (the tensor locating the center of the yield surface in stress space), ii) the parameters that characterize the expansion of the yield surface, iii) the parameters that characterize the 'boundary-surface' in multi-yield-surface theories of plasticity [3-7], and iv) the 'drag-stres' used to characterize the creep surface.

Here, the concept of intrinsic time dependent on plastic strain is used for the derivation of the differential or incremental form of the integral relation of stress and strain for plasticity. This derivation presents the endochronic theory in a structure that is similar to that of classical plasticity, thus, leading to a stiffness type finite finite element formulation.

While the endochronic relation as developed by Watanabe & Atluri [8] is similar in its structure to that of classical plasticity, there are several novel advantages present in the endochronic theory not present in the classical plasticity theory. The ability to model test data for both

monotonic or cyclic plasticity as accurately as possible, with a minimal number of material parameters makes the endochronic theory a simple theory to implement in a finite element code.

Following, is a summarized table of the rate form of the endochronic theory. The detail of the derivation will be shown in Appendix B.

Table 8.1

Summary of the Internal-Time Theory of Plasticity

Endochronic Theory:

$$d\sigma_{kk} = (2\mu+3\lambda)\, d\varepsilon_{kk}(\underset{\sim}{\varepsilon})$$

Where μ, λ are lame constants

$$f(\zeta) = 1 + a\zeta \qquad \text{(linear); or}$$

$$f(\zeta) = a + (1-a)e^{-\gamma\zeta} \qquad \text{(exponential)}$$

$$c = 1 + \rho_1(o) + \frac{(\underset{\sim}{S}-\underset{\sim}{\alpha}):\underset{\sim}{h}^*}{S_y^{\,o} f^2(\zeta)} + \frac{S_y^{\,o}(df/d\zeta)}{2\mu}$$

$$S_y^{\,o} = 2\mu o_o$$

$$\rho(z) \equiv \rho_o \delta(z) + \rho_1(z) \left(= \sum_i \rho_{1i}\, e^{-a_i z}\right)$$

$$\underset{\sim}{\alpha} = \sum_i \alpha_i^{(i)} \; ; \; \underset{\sim}{h}^* = \sum_i \underset{\sim}{h}^{*(i)} = \sum_i -\frac{a_i}{2\mu}\, \underset{\sim}{\alpha}^{(i)}$$

Rate of Kinematic Hardening:

$$d\underset{\sim}{\alpha}^{(i)} = 2\mu\, \rho_{1i}\, d\underset{\sim}{\varepsilon}^p - \frac{a_i \underset{\sim}{\alpha}^{(i)}}{f} \left(d\underset{\sim}{e}^p : d\underset{\sim}{\varepsilon}^p\right)^{\frac{1}{2}}$$

(no sum on i) for i = 1,2.....

$$d\underset{\sim}{\alpha} = \sum_i d\underset{\sim}{\alpha}_i^{(i)} = 2\mu\, \rho_{1i}\, d\underset{\sim}{e}^p - \sum_i \frac{a_i \underset{\sim}{\alpha}^{(i)}}{f} \left(d\underset{\sim}{\varepsilon}^p : d\underset{\sim}{\varepsilon}^p\right)^{\frac{1}{2}}$$

Table 8.1

Summary of the Internal-Time Theory of Plasticity
(Continued)

Rate of Isotropic Hardening:

(linear f)

$$dS = S_y^0 \, \beta \, (d\underset{\sim}{e}^P : d\underset{\sim}{e}^P)^{\frac{1}{2}}$$

(exponential f)

$$= r \left\{ S_y^\infty - S_y^0 \left[a + (1-a)e^{-r\zeta} \right] \right\} (d\underset{\sim}{e}^P : d\underset{\sim}{e}^P)^{\frac{1}{2}}$$

8.2.5 Residual Calculation and Iteration Scheme

Assuming that the material at state n+1 involves plastic process, then the resulting stress field will not be in equilibrium; however, one may compute the out-of-balance loads needed to produce an equilibrated stress field at that state. For equilibrium at state n+1, one should have:

$$\sigma_{ij,j}^{n+1} + f_i^{n+1} = 0 \qquad (8.28a)$$

$$\sigma_{ij}^{n+1} \, n_j = T_i^{n+1} \qquad (8.28b)$$

Weighing the above with the test function δu_i will give after application of Green's theorem:

$$R_{\sigma_k} \delta q_k = \int_{V_m} \sigma_{ij}^{n+1} \, \psi_{ik,j} \, \delta q_k \, dV \qquad (8.29)$$

The out-of-balance loads will be:

$$Q_k^{(i)} = \int T_i^{n+1} \, \psi_{ik} \, dS + \int f_i^{n+1} \, \psi_{ik} \, dV - R_{\sigma_k}^{(i)} \qquad (8.30)$$

For the points where the elastically applied stress exceeded the yield stress, the process should be plastic. The stiffness matrix may be updated to reflect the plastic process and allow the out-of-balance loads to follow the plastic

stress-strain path. This will give a correction to the displacements as well as the stress. However, when the strain is computed from the stress through the constitutive relations, there will be error in compatibility.

To enforce this condition, a weighting function of the same class as the function for the stress field may be used. The following residual load due to the error in compatibility is obtained:

$$\underset{\sim}{R}_{\varepsilon}\delta\underline{\beta} = \int_{V_m} \left\{ \varepsilon_{ij}^{n+1} - \frac{1}{2}\left(u_{i,j} + u_{j,i}\right)^{n+1} \right\} \delta\, \sigma_{ij}\, dV \tag{8.31}$$

Application of the above residual to the system will give displacements that are compatible with the strains obtained from the stress field. There will be some redistribution of the stress when the strain residual is applied, but for the most part, the displacements will change more during each iteration than the stress. One may apply both $\underset{\sim}{R}_{\sigma}$ and $\underset{\sim}{R}_{\varepsilon}$ at the same time, and continue the iteration process until the norm of the displacement does not change significantly.

8.2.6 Consistency Condition

With the above hybrid method, unless the stress/strain increments are very small, there will be errors in the consistency condition.

$$F^{n+1} \equiv \left(\underset{\sim}{\sigma}^{n+1} - \underset{\sim}{\alpha}^{n+1}\right) : \left(\underset{\sim}{\sigma}^{n+1} - \underset{\sim}{\alpha}^{n+1}\right) - R^2 = 0 \tag{8.32}$$

A mid-point rule is used for the integration of the strains to reduce the errors. Considering the finite change of $\Delta\underset{\sim}{\sigma}$, the plastic strain may be computed as:

$$\Delta\underset{\sim}{\varepsilon}^{p} = \underset{\sim}{N}^{*} \frac{\underset{\sim}{N}^{*} : \Delta\underset{\sim}{\sigma}}{2\mu(C-1)} \tag{8.33}$$

where

$$\underset{\sim}{N}^{*} = \frac{\left(\left(\underset{\sim}{\sigma}^{n} + \beta\Delta\underset{\sim}{\sigma}\right) - \left(\underset{\sim}{\alpha}^{n} + \beta\Delta\underset{\sim}{\alpha}\right)\right)}{\left\|\left(\underset{\sim}{\sigma}^{n} + \beta\Delta\underset{\sim}{\sigma}\right) - \left(\underset{\sim}{\alpha}^{n} + \beta\Delta\underset{\sim}{\alpha}\right)\right\|} \tag{8.34}$$

Likewise, the change in strain for a plastic process may be approximated as:

$$\Delta\underset{\sim}{\varepsilon} = \frac{\Delta\underset{\sim}{\sigma}}{2\mu} + \frac{\lambda}{(3\lambda+2\mu)2\mu}\left(\underset{\sim}{N}^{*} : \underset{\sim}{I}\right)\underset{\sim}{N}^{*} - \frac{\underset{\sim}{N}^{*}\,\underset{\sim}{N}^{*} : \Delta\underset{\sim}{\sigma}}{2\mu\,(C-1)} \tag{8.35}$$

Using the mid-point rule will lead to compatibility errors. However, the application of the residual $\underline{R}_\varepsilon$ will correct the errors that accumulate due to compatibility.

The final system of equations that result when both $\underline{R}_\varepsilon$ and $\underline{R}_\sigma$ are applied during the iteration process will have the form

$$\begin{pmatrix} -\underline{\underline{H}} & \underline{\underline{G}} \\ \underline{\underline{G}}^T & \underline{\underline{0}} \end{pmatrix} \begin{Bmatrix} \Delta\underline{s}^i \\ \Delta\underline{q}^i \end{Bmatrix} = \begin{Bmatrix} \underline{R}_\varepsilon^i \\ \underline{R}_\sigma^i + \Delta\underline{T} + \Delta\underline{F} \end{Bmatrix} \tag{8.36}$$

Here, the matrix $\underline{\underline{G}}$ is constant and only need be evaluated once. The matrix $\underline{\underline{H}}$ depends on the material behavior, and must be evaluated for each iteration.

As each iteration, i, is carried out, the stress and displacement are updated as:

$$\underset{\sim}{\sigma}^{n+1} = \underset{\sim}{\sigma}^n + \Delta\underset{\sim}{\sigma}_{AB} + \sum_i \Delta\underset{\sim}{\sigma}^i \tag{8.37a}$$

$$\underline{q}^{n+1} = \underline{q}^n + \Delta\underline{q} + \sum_i \Delta\underline{q}^i \tag{8.37b}$$

The strain must be computed in two parts with the elastic part given by

$$\Delta\underset{\sim}{\varepsilon}_{AB} = \underline{\underline{S}}^e \Delta\underset{\sim}{\sigma}_{AB} \tag{8.38}$$

and the plastic part by

$$\Delta\underset{\sim}{\varepsilon}_{BC} = \underline{\underline{S}}^e \Delta\underset{\sim}{\sigma}^i + \underline{\underline{S}}^p \Sigma_i \Delta\underset{\sim}{\sigma}^i \tag{8.39}$$

8.2.7 Implementation of the 16-Node Stress-Based Hybrid Element in NESSUS

The stresses within an element were represented as a summation of equilibrated polynomial stress modes, A_{ijm}, with undetermined parameters Δs_m;

$$\Delta\sigma_{ij} = A_{ijm}\,\Delta\beta_m \qquad (8.40)$$

Following the guidance and suggestions from Punch and Atluri [2], 42 stress modes were selected. The details of the derivation and the exact form of the polynomials used are presented in Appendix B. To enforce compatibility, the test functions $\delta\sigma_{ij}$ chosen are of the same class as in the trial functions for stress.

The displacements are interpolated from the nodal values, and the standard isoparametric shape functions are used. The exact form of shape functions is shown in Appendix B.

Once the trial and test functions for stresses and displacements are determined, the needed matrices $\underline{H}$, $\underline{G}$, $\Delta\underline{T}$, and $\Delta\underline{F}$ can be evaluated. A flow chart is presented in Appendix B to show the complete procedure.

8.2.8 Validation Problems

1. Linear Case

Introduction

A standard test problem for finite elements applied to the field of shell analysis is the pinched cylinder problem. It was carried out by Cantin and Clough [9] with a special displacement based cylindrically curved element. Henshell et al. [10] used an assumed stress hybrid element with both conforming and non-conforming versions. Later Ashwell and Sabir [11] analyzed this problem by using a cylindrical shell element which is based on independent strain functions. Various mesh sizes were used by these works and convergence results were reported elsewhere.

Results

The dimensions of this pinched cylinder are shown in Figure 8.1. Due to the symmetric behavior of the geometry and loading, only one eighth of the domain is modeled. Two thickness values (0.094 in. and 0.01548 in.) are used to simulate thick and thin shell structure respectively.

For the thick cylinder (t=0.094 in), an inextensional theory was used by Timoshenko and Woinowsky-Krieger [12] and the deflection of 0.1084 in. was reported which is known to be too low. Cantin and Clough [13] obtained the value of 0.1128 in. by dividing the octant of the cylinder into three elements longitudinally and 49 circumferentially, with 1200 degrees of freedom.

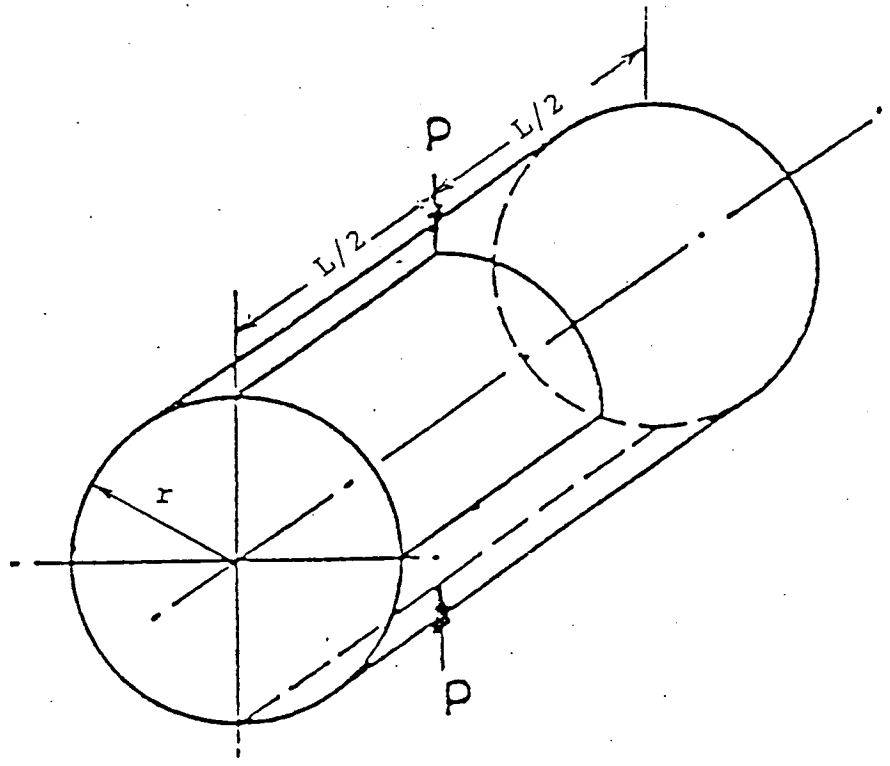


Figure 8.1 Pinched Cylinder Problem

L = 10.35 in., r = 4.953 in.

E = 10.5x10E6 lbf/in.^{2}, ν = 0.3125

P = 100 lbf

Thickness = 0.094 in. (thick), 0.01548 in. (thin).

Henshell et al. [10] used the assumed stress hybrid element with mesh sizes from 1x1 up to 6x6 and reached a converged displacement as 0.118 in. For the present 16-node stress-based hybrid element, two meshes (1x4 and 1x8) were used and the displacement of 0.118 in. was obtained from both meshes as shown in Table 8.2.

For the thin cylinder (t=0.01548 in.), the approximated analytical solution 0.2439 in. was compared with the present hybrid element, with three different mesh sizes (1x4, 1x8, and 2x8), as shown in Table 8.2. Also listed in Table 8.3 is the displacement solution obtained by Ashwell and Sabir [1] with the cylindrical shell element based on strain functions.

It is clear that the solution based on the present 16-node stress-based hybrid element converges for both thick and thin shell problems, and agrees well with other numerical solutions.

2. Non-linear Case

Introduction

One of the popular problems in the field of elasto-plastic structural analysis is that of a perforated plate under tension loading. Extensive works have been carried out in prior literature by using experimental testing and finite element techniques. Theocaris and Marketos [13] handled this problem experimentally by using photo-elastic coating techniques. Total strains and plastic strains were reported as well as stresses which are estimated by Prandtl-Reuss incremental plasticity relations. Finite element methods were used by many researchers, among them Marcal and King [14]; Yamada, Yoshimura, and Sakurai [15]; Zienkiewicz, Vallippan, and King [16]; and Bartelds [17]. Though the problem was analyzed by researchers for both cases of plasticity with and without strain hardening, only the case with strain hardening is studied here.

Results

The perforated plate problem considers a plate with a center hole under uniform tension as shown in Figure 8.2. Due to the symmetric characteristics of the geometry and loading, only one eighth of the plate is modeled.

Increments of load equal to 0.1 of the load at first yield were used. The plastic zones at these loading steps are presented in Figure 8.3, which are in good agreement with results obtained by Zienkiewicz et al., who use constant strain triangle with an 'initial stress' finite element approach.

Table 8.2

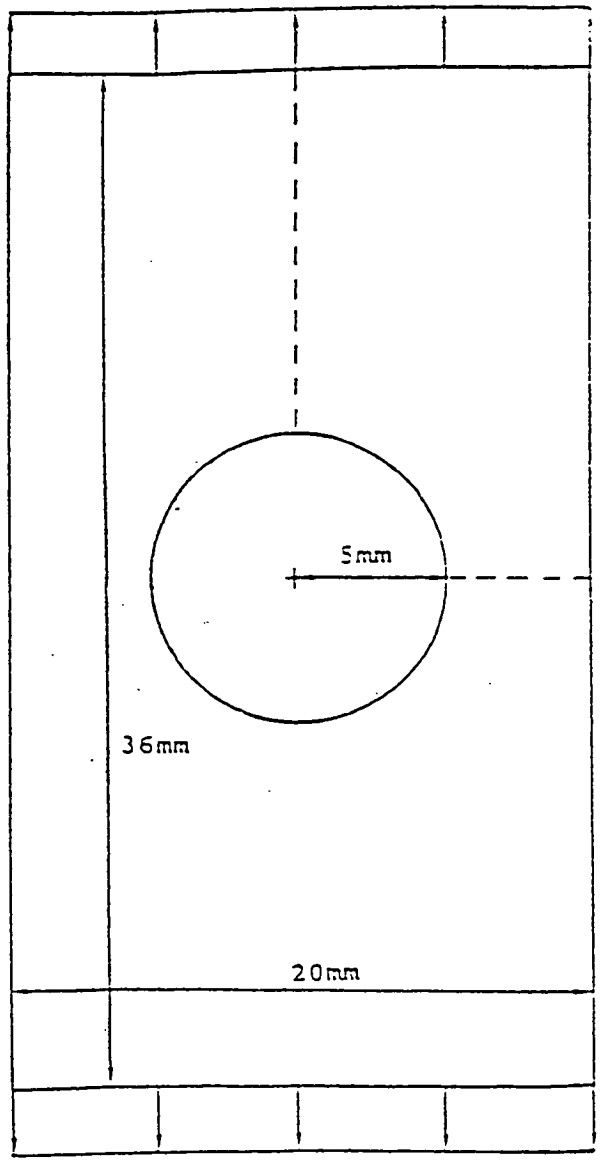
Deflection under One Load for Thick Pinched Cylinder Problem
(t=0.094 in.)

Present		Henshell et al		Cantin & Clough	
mesh	disp. (in.)	mesh	disp. (in.)	mesh	disp. (in.)
1 X 4	0.11840	1 X 1	0.1166	1 X 3	0.0297
1 X 8	0.11813	2 X 2	0.1111	2 X 5	0.0780
		3 X 3	0.1049	1 X 7	0.0987
		4 X 4	0.1170	2 X 7	0.1002
		5 X 5	0.1173	3 X 49	0.1128
		6 X 6	0.1174		

Table 8.3

Deflection Under One Load for Thin Pinched Cylinder Problem
(t=0.01546 in.)

Analytical	Present		Ashwell & Sabir	
disp. (in.)	mesh	disp. (in.)	mesh	disp. (in.)
0.02439	1 X 4	0.023516	1 X 4	0.02403
	1 x 8	0.024891	1 x 8	0.02406
	2 x 8	0.024315	2 x 4	0.02409
			2 X 8	0.02414
			3 x 4	0.02414
			3 x 8	0.02418
			8 x 8	0.02431



Aluminum alloy 57S

$E = 7000 \text{ kg/mm}$

$\nu = 0.2$

$\bar{\sigma}_y = 24.3 \text{ kg/mm}$

$\rho_o = S_y^o/2\mu = \left(\sqrt{\frac{2}{3}}\,\bar{\sigma}_y\right)/2\mu$

$= 3\ 40129 \times 10^{-3}$

$H_e' = \frac{3}{2} f'(\rho) = \frac{1}{S_y^o} \frac{d\bar{\sigma}}{d\bar{\epsilon}^p}$

$= 11.28983$

Figure 8.2 Perforated Tension Strip

$$\text{Load Factor } \mu = \frac{\sigma_{mean}}{\sigma_y}$$

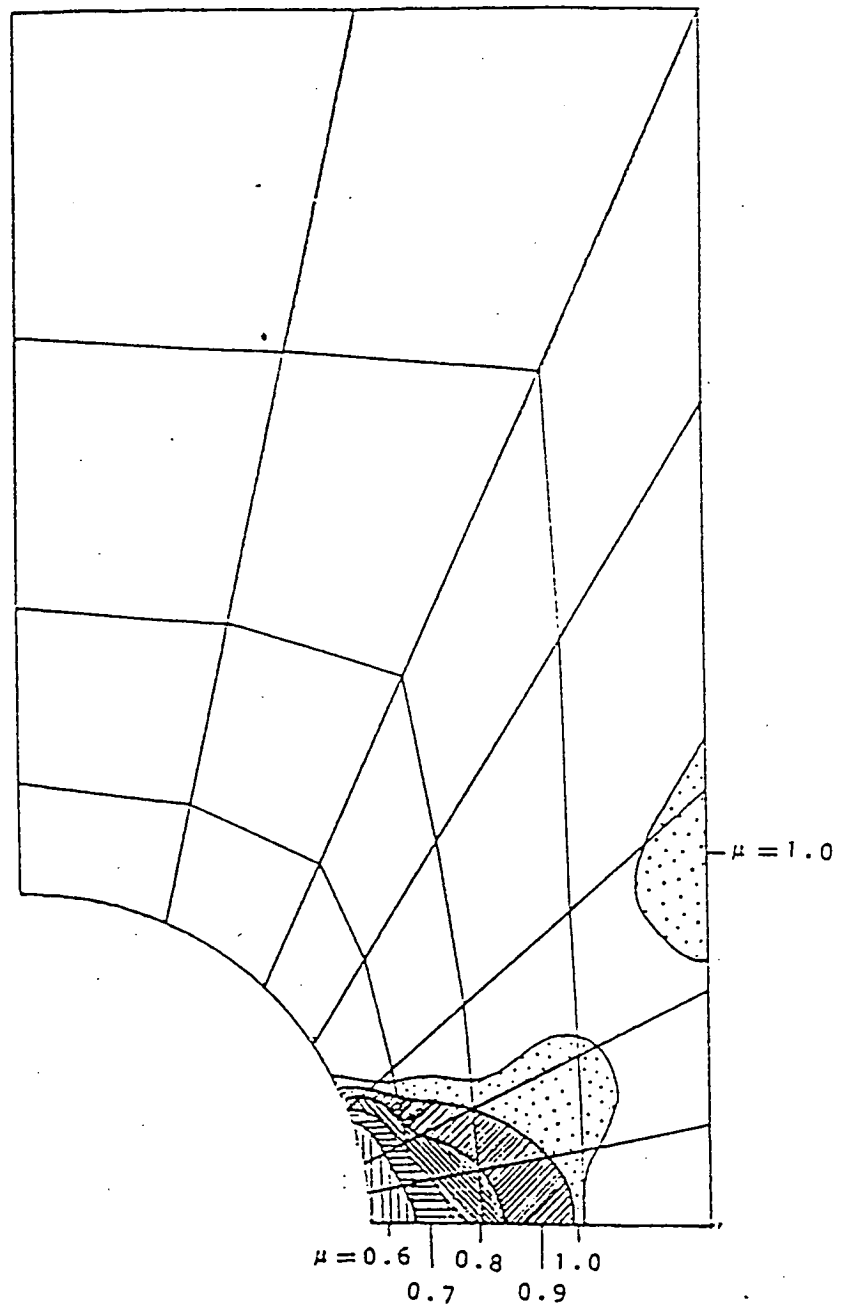


Figure 8.3 Finite Element Mesh and Plastic Zones at Load Factors μ = 0.6,0.7,0.8,0.9,1.0

The calculated total strains are compared with the experimental results from Theokaris and Marketos [13] and other finite element solutions [14,16] as shown in Figure 8.4. It can be seen that the solutions from present method agree well with others.

8.2.9 Consideration of Low to Zero Strain Hardening Problem

The present stress-based hybrid formulation provides a more accurate representation of the stress and the strain than the displacement based method, yet the formulation breaks down for certain classes of material behavior. For elastic perfectly plastic material, or for a non-linear hardening material where the tangent modulus becomes very small, the stress based method, in the present variational formulation, is incapable of correctly modeling the solid. Likewise, for elastically incompressible materials, the stress based method, in the present form, breaks down.

The magnitude of plastic strain in the endochronic theory was expressed in terms of the strain increment as

$$d\zeta = \frac{1}{C} \underset{\sim}{N} : d\underset{\sim}{\varepsilon}' \tag{8.41}$$

To express the magnitude of plastic strain in terms of the stress increment, one may, through, a simple substitution, note

$$d\zeta = \frac{1}{C} \underset{\sim}{N} : \left(\frac{d\underset{\sim}{S}}{2\mu} + d\underset{\sim}{\varepsilon}'\right) \tag{8.42}$$

$$= \frac{1}{C} \underset{\sim}{N} : \frac{d\underset{\sim}{S}}{2\mu} + \frac{d\zeta}{C} \tag{8.43}$$

or

$$d\zeta = \frac{1}{(C-1)2\mu} \left(\underset{\sim}{N} : d\underset{\sim}{S}\right) \tag{8.44}$$

which gives the plastic strain as

$$d\underset{\sim}{\varepsilon}^p = \frac{1}{(C-1)2\mu} \left(\underset{\sim}{N} : d\underset{\sim}{S}\right) \underset{\sim}{N} \tag{8.45}$$

If C=1.0, then for the plastic strain to remain finite the stress increment must be orthogonal to the normal of the yield surface,

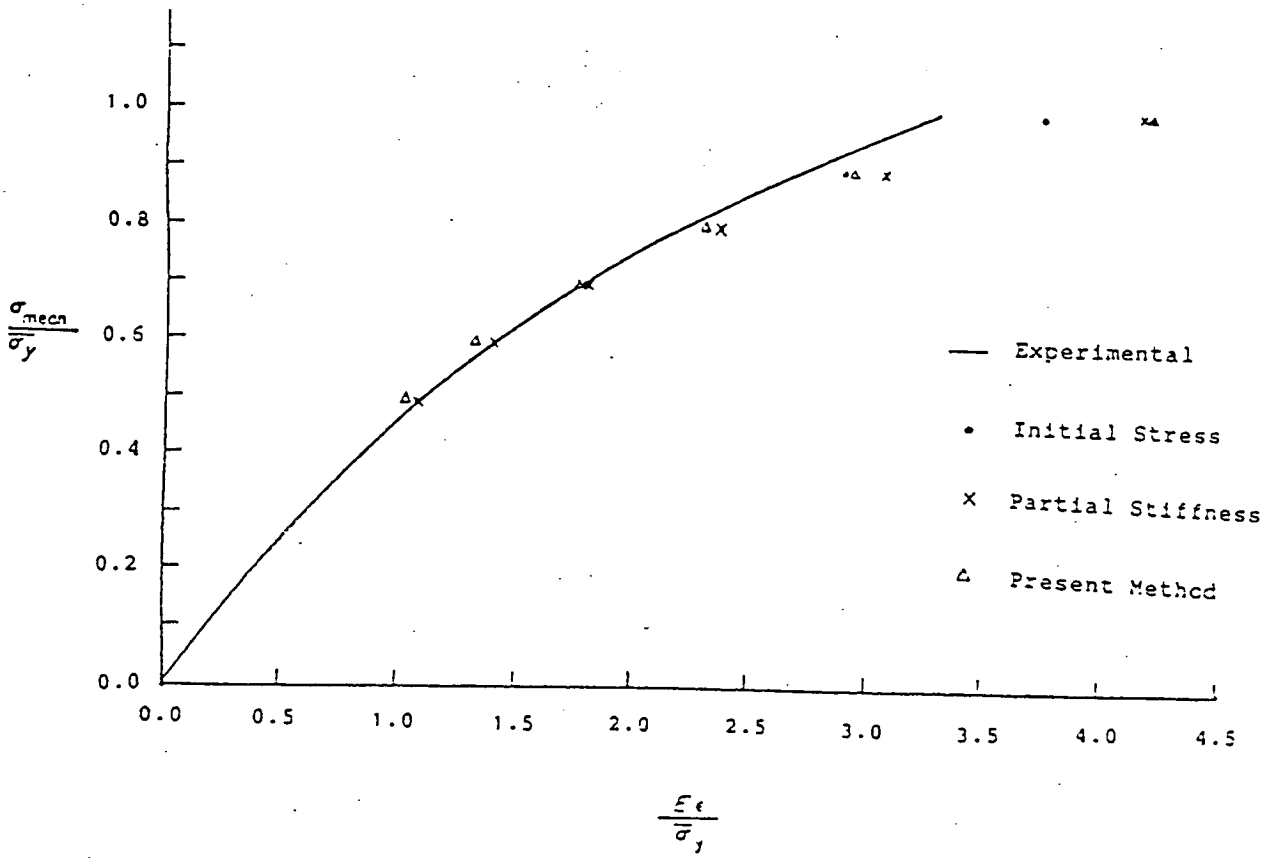


Figure 8.4 Development of Maximum Strain Point of First Yield

$$\underset{\sim}{N} : d\underset{\sim}{S} = 0 \tag{8.46}$$

In the limit as C tends to 1.0, the hybrid finite element formulation will not be feasible due to the inability of a computer to numerically evaluate the limit of

$$\lim_{C \to o} \frac{N_{ij} dS_{ij}}{(C-1)} \tag{8.47}$$

For values of C that are close to 1.0, as is the case when a material is in the range of strain hardening where the rate of hardening is such that the stress strain curve is nearly horizontal, numerical difficulties will occur.

In order to avoid numerical difficulties when C is approximately 1.0, the variational statement should be reformulated by introducing a new field variable for the magnitude of plastic strain, $d\zeta$. The magnitude of plastic strain is related to the deviatoric stress increment through

$$d\zeta = \frac{1}{(C-1)2\mu} \underset{\sim}{N} : d\underset{\sim}{S} \tag{8.48}$$

The total strain is given by

$$d\underset{\sim}{\varepsilon} = d\underset{\sim}{\varepsilon}^e + d\underset{\sim}{\varepsilon}^p \equiv d\underset{\sim}{\varepsilon}^e + d\zeta \underset{\sim}{N} \tag{8.49}$$

where the increment of plastic strain is expressed in terms of a normal to the yield surface with magnitude $d\zeta$. Expressed this way, the compatibility equation becomes, assuming that the constitutive equation is satisfied a priori,

$$du_{i,j} = \frac{d\underset{\sim}{\sigma}}{2\mu} + \frac{\lambda}{(3\lambda + 2\mu)2\mu} (d\underset{\sim}{\sigma} : \underset{\sim}{I})\underset{\sim}{I} + d\zeta \underset{\sim}{N} \tag{8.50a}$$

or

$$du_{i,j} = S^e_{ijk\ell} \, d\sigma_{k\ell} + d\zeta \, N_{ij} \tag{8.50b}$$

with

$$d\zeta(C-1)\,2\mu = (\underset{\sim}{N} : d\underset{\sim}{S}) \qquad (8.51)$$

$$= (\underset{\sim}{N} : d\underset{\sim}{\sigma})$$

When C=1.0, the stress increment must be orthogonal to the yield surface, and the magnitude of plastic strain is no longer coupled to the trace of the stress increment with the normal to the yield surface. Instead, the magnitude of plastic strain is determined from the compatibility condition. The magnitude of plastic strain will become an undetermined parameter to be resolved at the global level.

8.2.10 Weak Form

For the incremental formulation, the finite change in magnitude of plastic strain, $\Delta\zeta$ must be considered. The weak form for the magnitude of plastic strain is expressed by using a weighting function Δh of the same class as $\Delta\zeta$. The weak form becomes:

$$\sum_m \int_{V_m} T \left\{\Delta\zeta\,(C-1)2\mu - \underset{\sim}{N} : \Delta\underset{\sim}{\sigma}\right\} d\zeta\, dV = 0 \qquad (8.52)$$

The compatibility condition may still be expressed in weak form through the use of the weighting function $\delta\sigma_{ij}$. The weak form of the compatibility condition becomes:

$$\sum_m \int_{V_m} \left\{S^e_{ijk\ell}\,\Delta\sigma_{k\ell} + T\,\Delta\zeta\,N_{ij} - \Delta U_{ij}\right\} \delta\sigma_{ij}\, dV = 0 \qquad (8.53)$$

Applying Green's theorem, the combined weak form may be expressed by combining the weak form for compatibility and plastic strain with the weak form for equilibrium (8.16), and traction boundary condition (8.5,8.7) as:

$$\sum_m \int_{V_m} - S^e_{ijk\ell}\Delta\sigma_{k\ell}\delta\sigma_{ij}dV + \int_{V_m} \Delta u_{i,j}\,\delta\sigma_{ij}\,dV$$

$$- \int_{V_m} \Delta\zeta\,N_{ij}\,\delta\sigma_{ij}\,dV + \int_{V_m} \delta u_{i,j}\Delta\sigma_{ij}\,dV + \int_{S_\sigma} \Delta T_i\,\delta u_i\,dS \qquad (8.54)$$

$$+ \int_{V_m} \Delta f_i\,\delta u_i\,dV + \int \Delta\zeta\left((C-1)2\mu - N_{ij}\Delta\sigma_{ij}\right) d\zeta\, dV = 0$$

8.2.11 Discrete Weak Form

In the stress-based method, the stress increment, $\Delta\sigma_{ij}$, was defined through a set of polynomial basis function, A_{ijn}, and undetermined stress parameters $\delta\beta_n$. Likewise, the change in magnitude to plastic strain $\Delta\zeta$, was defined through a set of polynomial bases functions, Λ_k, and undetermined parameters $\Delta\alpha_k$. The test function, $\delta\zeta$, may be defined through the same basis functions.

$$\Delta\zeta = \Delta\alpha_k \Lambda_k$$
$$\delta\zeta = \delta\alpha_k \Lambda_k \tag{8.55}$$

Using the same basis functions for the stress and displacement, as was used in the hybrid method, a three field variational statement was formulated. After substitution of

$$\Delta\zeta = \Delta\alpha_k \Lambda_k, \quad \Delta\sigma_{ij} = A_{ijn}\Delta\beta_n, \quad \Delta u_n = \psi_{n\ell}\Delta q_\ell \tag{8.56}$$
$$\delta\zeta = \delta\alpha_k \Lambda_k, \quad \delta\sigma_{ij} = A_{ijn}\delta\beta_n, \quad \delta u_n = \psi_{n\ell}\delta q_\ell$$

and defining the matrices

$$\underline{\underline{H}}_{\beta\beta} = \int_{V_m} \Lambda_{ijm} S_{ijk\ell} \Lambda_{k\ell n} \, dV \tag{8.57a}$$

$$\underline{\underline{H}}_{\alpha\beta} = \int_{V_m} T \Lambda_k N_{ij} A_{ijn} \, dV \tag{8.57b}$$

$$\underline{\underline{H}}_{\alpha\alpha} = \int_{V_m} \Lambda_k 2\mu(C-1) T \Lambda_\ell \, dV \tag{8.57c}$$

$$\underline{\underline{G}} = \int_{V_m} \psi_{i\ell,j} A_{ijn} \, dV \tag{8.57d}$$

where

$$N_{ij} = \frac{\sigma_{ij} - \alpha_{ij}}{R} \tag{8.58}$$

The combined weak form may be expressed as:

$$- \delta\underline{\beta}^T \underline{\underline{H}}_{\beta\beta} \Delta\underline{\beta} + \delta\underline{\beta}^T \underline{\underline{G}} \Delta\underline{q} - \delta\underline{\beta}^T \underline{\underline{H}}_{\alpha\beta} \Delta\underline{\alpha} - \delta\underline{\alpha}^T \underline{\underline{H}}_{\alpha\alpha} \Delta\underline{\alpha} + \tag{8.59}$$

$$\delta\underline{\alpha}^T \underline{\underline{H}}_{\alpha\beta} \Delta\underline{\beta} + \Delta\underline{\beta}^T \underline{\underline{G}} \delta\underline{q} = \Delta\underline{T}^T \delta\underline{q} + \Delta\underline{F}^T \delta\underline{q}$$

The following system of algebraic equations results for arbitrary $\delta\underline{\alpha}$, $\delta\underline{\beta}$ and $\delta\underline{q}$.

$$\begin{bmatrix} -\underline{\underline{H}}_{\beta\beta} & -\underline{\underline{H}}_{\alpha\beta} & \underline{\underline{G}} \\ -\underline{\underline{H}}_{\alpha\beta} & \underline{\underline{H}}_{\alpha\alpha} & \underline{\underline{0}} \\ \underline{\underline{G}}^T & \underline{\underline{0}} & \underline{\underline{0}} \end{bmatrix} \begin{Bmatrix} \Delta\underline{\beta} \\ \Delta\underline{\alpha} \\ \Delta\underline{q} \end{Bmatrix} = \begin{Bmatrix} \underline{0} \\ \underline{0} \\ \Delta\underline{T} + \Delta\underline{F} \end{Bmatrix} \tag{8.60}$$

As in the stress-based hybrid method, the parameters $\Delta\underline{\alpha}$ and $\Delta\underline{\beta}$ may be eliminated at the element level. This reduction is possible because the parameters for stress and magnitude of plastic strain were defined in such a way that no coupling occurred between elements. When $\underline{\underline{H}}_{\alpha\alpha}$ is non-singular one may express $\Delta\underline{\alpha}$ in terms of $\Delta\underline{\beta}$ as

$$\Delta\underline{\alpha} = \underline{\underline{H}}_{\alpha\alpha}^{-1} \underline{\underline{H}}_{\alpha\beta} \Delta\underline{\beta} \tag{8.61}$$

Substitution then allows $\Delta\underline{\beta}$ to be expressed in terms of the displacement as

$$\left(-\underline{\underline{H}}_{\beta\beta} - \underline{\underline{H}}_{\alpha\beta} \underline{\underline{H}}_{\alpha\alpha}^{-1} \underline{\underline{H}}_{\alpha\beta}\right) \Delta\underline{\beta} + \underline{\underline{G}} \Delta\underline{q} = \underline{0} \tag{8.62}$$

or

$$-\underline{\underline{H}}^{*} \Delta\underline{\beta} + \underline{\underline{G}} \Delta\underline{q} = \underline{0} \tag{8.63}$$

For non-singular $\underline{\underline{H}}^{*}$ the usual hybrid stiffness matrix results and may be assembled and solved as a system of algebraic equations.

$$\sum_{m} \underline{\underline{G}}^{T} \underline{\underline{H}}^{*-1} \underline{\underline{G}} \, \Delta\underline{\alpha} = \Delta\underline{T} + \Delta\underline{F} \tag{8.64}$$

When the material is in the fully plastic range, the orthogonality constraint prevents the inversion of $\underline{\underline{H}}_{\alpha\alpha}$ due to its singular form.

In the event that $\underline{\underline{H}}_{\alpha\alpha}$ is singular, $\Delta\underline{\alpha}$ is retained as a global variable and solved for as an unknown parameters at the global level. While for a single element $\underline{\underline{H}}_{\alpha\alpha}$ may be singular, the global system of equations will be non-singular if proper boundary conditions are applied. The number of stress parameters at the global level may still be reduced provided $\underline{\underline{H}}_{\beta\beta}$ is non-singular.

Since the incompressibility constraint arose as a limit condition, all elements would not be expected to behave in an incompressible manner. A substantial savings can be made if the decision to reduce or retain the parameters for the magnitude of plastic strain is made for each element. The criteria used to determine if an element should be reduced may be based upon the value of C at each Gauss point in the element.

By setting up the element variational statement for an element, as in equation (8.60), the matrix may be partitioned into reducible degrees of freedom and non-reducible degrees of freedom. Employing a standard substructure algorithm [18,19] allows one to reduce the unnecessary parameters while mapping the required parameters and reduced stiffness terms to global. Likewise, once the global parameters have been determined, the back substitution to find the reduced degrees of freedom may be implemented through a standard substructure algorithm.

8.2.12 Considerations for Finite Deformation

Rigorous and consistent formulations for numerical analysis of elastic-plastic large strain problems have become necessary due to the increased importance in recent years of analyzing problems such as metal forming processes, ductile fracture initiation and stable crack growth in cracked bodies, etc. Indeed several such formulations, and applications of the same, have appeared in recent literature. Among these can be cited the works of: Hibbit, Marcal and Rice [20], who use a total Lagrangean (TL)

formulation; Needleman and Tvergaard [21,22] and Hutchinson [23], who also use a TL formulation using convected coordinates; Yamada et al. [24] who use an updated Lagrangean (UL) formulation; Oasias [25], who uses an UL scheme which, due to the use of an elastic-plastic rate constitutive law that does not admit to a potential, leads to non-symmetric stiffnesses through a Galerkin scheme: McMeeking and Rice [26], who also use an UL scheme, which through the use of a rate constitutive law with a potential leads to symmetric stiffnesses; and Nemat-Nasser and Taya [27], whose formulation represents a modification of that in Ref. [26] to improve the accuracy in the case of large deformation of compressible materials. All of these rate formulations [20-27] are based on the principle of virtual work, or its variant, a variational principle due to Hill [28]. Thus, all the above works are based on assumed displacement type finite element methods.

A stress-based hybrid formulation for the analysis of finite deformation problems was presented by Atluri [29] at early 1970's. Later a series of research works on using hybrid formulation based on complementary energy principle or its rate form were done by Atluri and his colleagues [30-63]. The problem of determining suitable stress modes for hybrid or mixed formulations in the finite strain analysis has also been investigated by Atluri etc., and guidances and recommendations to choose those stress modes which will result in stable, invariant and least order elements were reported [64-67]. The endochronic theory which has its superior constitutive modeling capability, in cross-hardening, cyclic hardening and initial strain problems, over the classical theory has been successfully implemented into hybrid finite element method for finite deformation analysis by Atluri et al. [68-77].

For finite deformation analysis, a rate form stress-based hybrid formulation can be found in [37]. This formulation, based on the Hellinger-Reissner principle with total Lagrangean approach, can be implemented into any existing finite element program.

The weak form of this stress-based hybrid formulation can be represented as:

$$\delta\{\sum_m \int_V [\ \frac{1}{2} S_{ik}\Delta u_{j,k}\Delta u_{j,i} + \frac{1}{2} S_{ij}\ (\Delta u_{i,j}+\Delta u_{j,i}+\Delta u_{k,i}\Delta u_{k,j})$$

$$- \frac{1}{2} C_{ijkl}\ \Delta S_{ij}\ \Delta S_{kl}]\ dV\} \tag{8.65}$$

$$= \delta\ \{\sum_m \int_V \rho_o^{N+1} b_j \Delta u_j\ dV + \int_{S_\sigma} {}^{N+1}t_j \Delta u_j\ dS - \int_V\ (S_{ik}F_{jk}\Delta u_{j,i})\ dV$$

$$+ \int_V [\gamma_{ij}^N - \frac{1}{2}\ ({}^N u_{i,j} + {}^N u_{k,i}\ {}^N u_{kj})]\ \Delta S_{ij}\ dV\}$$

where

S_{ij} : second Kitchhoff stress

ΔS_{ij} : incremental second Kirchhoff stress

γ_{ij}^N : Green-Lagrange strain tensor at state N (8.66)

F_{ij} : deformation gradient tensor

b_j : body force per unit mass

The incremental second Kirchhoff stresses within an element were represented as a summation of polynomial stress modes, A_{ijm}, with undetermined parameters Δs_m;

$$\Delta S_{ij} = A_{ijm}\ s_m \tag{8.67}$$

The displacements were interpolated from the nodal values, Δq_k, and the standard isoparametric shape functions, ψ_{ik} as:

$$\Delta u_i = \psi_{ik}\Delta q_k \tag{8.68}$$

The same form of polynomial stress modes and interpolation functions will be used for the test functions δS_{ij} and δu_i, respectively.

Substitute these discrete forms into the weak form (8.65) and rearrange terms, the following matrix form can be obtained,

$$- \underline{\beta}^T \underline{\underline{H}}\, \delta\underline{\beta} + \Delta\underline{q}^T \underline{\underline{K}}\, \delta\Delta\underline{q} + \delta\underline{\beta}^T \underline{\underline{G}}\, \Delta\underline{c} + \underline{\beta}^T \underline{\underline{G}}\, S\Delta\underline{q}$$

$$= \delta\Delta\underline{q}^T \underline{R}_e + \delta\underline{\beta}^T \underline{R}_c + \delta\, \Delta\underline{q}^T \Delta\underline{Q} \tag{8.69}$$

where

$$\underline{\underline{H}} = \int_V C_{ijk\ell} A_{ijm} A_{k\ell n}\, dV$$

$$\underline{\underline{K}} = \int_V S_{ik} \psi_{j\ell,k} \psi_{jn,i}\, dV$$

$$\underline{\underline{G}} = \int_V A_{ijr}(\psi_{ik,j} + \psi_{jk,i} + \psi_{kn,i}\psi_{kn,j})\, dV \tag{8.70}$$

$$\underline{R}_e = \int_V S_{ik} F_{jk} \psi_{jk,i}\, dV$$

$$\underline{R}_c = \int_V [\gamma_{ij}^N - \frac{1}{2}({}^N u_{i,j} + {}^N u_{j,i} + {}^N u_{ki}\, {}^N u_{k,j})] A_{ijm}\, dV$$

$$\Delta\underline{Q} = \int_V \rho_o {}^{N+1} b_j \psi_{jk}\, dV + \int_{S_\sigma} {}^{N+1} t_j \psi_{jk}\, dS$$

For arbitrary $\delta\underline{\beta}$, one must satisfy

$$- \underline{\underline{H}}\, \underline{\beta} + \underline{\underline{G}}\, \Delta\underline{c} = \underline{R}_c \tag{8.71}$$

Thus, $\underline{\beta}$ may be expressed in terms of displacement for the element as

$$\underline{\beta} = \underline{\underline{H}}^{-1}(\underline{\underline{G}}\, \Delta\underline{c} - \underline{R}_c) \tag{8.72}$$

For arbitrary $\delta\Delta\underline{q}$, then one must have at global level

$$\sum_m (\underline{\underline{K}}\,\Delta\underline{q} + \underline{\underline{G}}^T\underline{\beta}) = \sum_m (\underline{R}_e + \Delta\underline{Q}) \tag{8.73}$$

or

$$\sum_m (\underline{\underline{K}} + \underline{\underline{G}}^T\,\underline{\underline{H}}^{-1}\,G)\,\Delta\underline{q} = \sum_m (\underline{\underline{G}}^T\,\underline{\underline{H}}^{-1}\,\underline{R}_c + \underline{R}_e + \Delta\underline{Q}) \tag{8.74}$$

Then the final solutions can be obtained by the similar iterative scheme used for the elasto-plastic analysis presented in previous sections.

8.2.13 Considerations for Thermal Loading

In general, for structural analysis with large temperature change, the thermal effect is important and can not be ignored. The stress-based hybrid formulation presented in previous sections includes mechanical loading only. However, this formulation can easily be extended to account for thermal loading as well. The necessary modifications for the consideration of thermal loading are present as follows.

With the consideration of thermal loading, the total strain components can be separated into three parts,

$$\varepsilon_{ij} = \varepsilon_{ij}^{e} + \varepsilon_{ij}^{t} + \varepsilon_{ij}^{p} \tag{8.75}$$

where ε_{ij}^{e} and ε_{ij}^{p} are elastic and plastic strain components due to mechanical loading respectively and ε_{ij}^{t} are thermal strains which can be shown as

$$\varepsilon_{ij}^{t} = \delta_{ij}\kappa(T - T_o) \tag{8.76}$$

where δ_{ij} is Kronecker delta, κ is the linear coefficient of thermal expansion of the material, T is the prescribed temperature and T_o is the reference temperature.

Then the new modified weak form is

$$\sum_{m} \int_{V_m} \left\{ \left(S_{ijk\ell} \Delta\sigma_{k\ell} - \frac{1}{2}(\Delta u_{i,j} + \Delta u_{j,i}) + \Delta\varepsilon_{ij}^{t} \right) \delta\sigma_{ij} \right\} dV$$

$$+ \int_{S_\sigma} (\Delta\sigma_{ij} n_j - \Delta T_i)\delta u_i dS + \int_{S_\sigma} \Delta\sigma_{ij} n_j \delta u_i \, dS \qquad (8.77)$$

$$+ \int_{V_m} (\Delta\sigma_{ij,j} + \Delta f_i)\delta u_i \, dV = 0$$

After substitutions of the discreted form of the trial and test functions as those used in section 8.2, one may express the new matrix form as:

$$\sum_{m} - \Delta\underline{\beta}^T \underline{\underline{H}} \, \delta\underline{\beta} + \Delta\underline{\alpha}^T \underline{\underline{G}} \, \delta\underline{\beta} + \delta\underline{\alpha}^T \underline{\underline{G}} \, \Delta\underline{\beta} = \Delta\underline{F}^{t^T} \delta\underline{\beta} + \Delta\underline{T}^T \delta\underline{q} + \Delta\underline{F}^T \delta\underline{c} \qquad (8.78)$$

where

$$\Delta\underline{F}^t = \int_V \Delta\varepsilon_{ij}^{t} A_{ijm} dV \qquad (8.79)$$

For arbitrary $\delta\beta$, we can get the form

$$- \underline{\underline{H}} \, \Delta\underline{\beta} + \underline{\underline{G}} \, \Delta\underline{q} = \Delta\underline{F}^t \qquad (8.80)$$

Thus, $\Delta\underline{\beta}$ may be expressed in terms of $\Delta\underline{q}$ as

$$\Delta\underline{\beta} = \underline{\underline{H}}^{-1}(\underline{\underline{G}}\Delta\underline{q} - \Delta\underline{F}^t) \qquad (8.81)$$

For arbitrary $\delta\underline{q}$, the final global form can be obtained as

$$\underline{\underline{G}}^T \Delta\underline{\beta} = \Delta\underline{T} + \Delta\underline{F} \qquad (8.82)$$

or

$$\underline{\underline{G}}^T \underline{\underline{H}}^{-1} \, \underline{\underline{G}} \, \Delta\underline{c} = \underline{\underline{G}}^T \underline{\underline{H}}^{-1} \Delta\underline{F}^t + \Delta\underline{T} + \Delta\underline{F} \qquad (8.83)$$

References

[1] Pian, T.H.H., and Tong, P., "Basis of Finite Element Methods for Solid Continua," Int. J. Num. Meth. Engng., 1, 1969, pp. 3-28.

[2] Punch, E.F., and Atluri, S.N., "Development and Testing of Stable, Invariant, Isoparametric Curvilinear 2- and 3-D Hybrid-Stress Elements," Comp. Meth. in Appl. Mech. Engrg., Vol. 47, pp. 331-356, 1984.

[3] Mroz, Z., "On the Description of Anisotropic Workhardening," J. Mech. Phys. Sol. Vol. 15, pp. 163, 1967.

[4] Mroz, Z., "An Attempt to Describe the Behavior of Metals Under Cyclic Loads Using a More General Workhardening Model," Acta Mech., Vol. 7, pp. 199, 1969.

[5] Dafalias, Y.F., and Popov, E.P., "A Model of Nonlinearly Hardening Materials for Complex Loading," Acta Mech., Vol. 21, pp. 173, 1975.

[6] Dafalias, Y.F., and Popov, E.P., "Plastic Internal Variables Formalism of Cyclic and Anisotropic," J. Appl. Mech., Vol. 43, pp. 645, 1976.

[7] Krieg, R.D., "A Practical Two Surface Plasticity Theory," J. Appl. Mech., Vol. 42, pp. 641, 1975.

[8] Watanabe, O., and Atluri, S.N., "Constitutive Modeling of Cyclic Plasticity and Creep, Using an Internal Time Concept," Technical Report, Center for the Advancement of Computational Mechanics, Georgia Tech., 1984.

[9] Cantin, G. and Clough, R.W., "A Curved, Cylindrical-Shell,Finite Element," AIAA J., Vol. 6, 6, pp. 1057, 1968.

[10] Henshell, R.D., Neale, B.K., and Warburton, G.B., "A New Hybrid Cylindrical Shell Finite Element," J. of Sound Vib., Vol. 16, 4, pp. 519-531, 1971.

[11] Ashwell, D. G. and Sabir, A.B., "A New Cylindrical Shell Finite Element Based on Simple Independent Strain Functions," Int. J. Mech. Sci., Vol. 14, pp. 171-189, 1972.

[12] Timoshenko, S. and Woinowsky-Krieger, S., Theory of Plates and Shells, 2nd ed., McGraw-Hill, New York, 1959.

[13] Theokaris, P.S. and Marketos, E., "Elastic-Plastic Analysis of Perforated Thin Strips of Strain-Hardening Materials," J. Mech. Phys. Sol., 12, pp. 377-390, 1964.

[14] Marcal, P.V. and King, I.P., "Elastic-Plastic Analysis of Two Dimensional Stress Systems by the Finite Element Method," Int. J. Mech. Sci., Vol. 9, pp. 143-155, 1967.

[15] Yamada, Y., Yoshimura, N., and Sakurai, T., "Plastic Stress-Strain Matrix and its Application for the Solution of Elastic-Plastic Problems by the Finite Element Method," Int. J. Mech. Sci., Vol. 10, pp. 343-354, 1968.

[16] Zienkiewicz, O.C., Vallippan, S., and King, I.P., "Elasto-Plastic Solutions of Engineering Problems "Initial Stress", Finite Element Approach," Int. J. for Num. Meth. Eng., Vol. 1, pp. 75-100, 1969.

[17] Bartelds, G., "The Application of the Finite Element Displacement Method to Problems of Elastoplastic Deformation," Contributions to the Theory of Aircraft Structures, Delft University Press, 1972.

[18] Wilson, E., and Dovey, H., Int. J. for Num. Meth. in Engrg., Vol. 1, 1976, pp. 1.

[19] Elwi, A.E. and Murray, D.W., "Skyline Algorithms for Multilevel Substructure Analysis," Int. J. for Num. Meth. in Engrg., Vol. 21, pp. 465-479, 1985.

[20] Hibbit, H.D., Marcal, P.V. and Rice, J.R., "A Finite Element Formulation for Problems of Large Strain and Large Displacement," Int. J. Solids and Struc., 6, pp. 1069, 1970.

[21] Needleman, A., "A Numerical Study of Necking in Circular Cylindrical Bars," J. Mech. and Phys. of Solids, 20, pp. 111, 1972.

[22] Needleman, A. and Tvergaard, V., "Necking and Biaxially Stretched Elastic-Plastic Circular Plates," J. Mech. and Phys. Solids, 25/3, pp. 159, 1977.

[23] Hutchinson, J.W., "Finite Strain Analysis of Elastic-Plastic Solids and Structures," in Numerical Solutions of Nonlinear Structural Problems (Ed. Hartung, R.F.), ASME, NY, pp. 17, 1973.

[24] Yamada, Y., Hirakawa, T. and Wifi, A.S., "Analysis of Large Deformation and Bifurcation in Plasticity Problems by the Finite Element Method," Proc. of Int. Conf. on Finite Elements in Nonlinear Solid and Structural Mechanics, Geilo, Norway, C08.1, 1977.

[25] Osias, J.R., "Finite Deformation of Elastic-Plastic Solids: The Example of Necking in Flat Tensile Bars," Ph. D. Thesis, Carnegie-Mellon U., 1972.

[26] McMeeking, R.M. and Rice, J.R., "Finite Element Formulations for Problems of Large Elastic-Plastic Deformation," Int. J. Solids and Struc., 11, pp. 601, 1975.

[27] Nemat-Nasser, S. and Taya, M., "Model Studies of Ductile Fracture-I Formulation," J. Franklin Inst., 302/5-6, pp. 463, 1976.

[28] Hill, R., "Some Basic Principles in the Mechanics of Solids Without a Natural Time," J. Mech. and Phys. Solids, 7, pp. 209, 1959.

[29] Atluri, S.N., "Finite Elasticity Solutions Using Hybrid Finite Elements Based on a Complementary Energy Principle," ASME Winter Annual Meeting, San Francisco, December 1978.

[30] Atluri, S.N., "Finite Element Methods for Finite Strain Plasticity Problems in Metalforming," Int. Conf. on Computational Methods in Nonlinear Mechanics (sponsored by NSF), Austin, Texas, March 1979.

[31] Atluri, S.N., "Finite Elasticity Solutions Using Hybrid Finite Elements Based on a Complementary Energy Principle II. Incompressible Materials", 1979 Joint Applied Mechanics, Fluid Conf., Niagara Falls, New York, June 1979.

[32] Atluri, S.N., "Finite Deformation Analysis of Shells", 24th AIAA/ASME/ASCE/AHS Struc, Structural Dynamics and Materials Conf., Lake Tahoe, Nevada, May 1983.

[33] Atluri, S.N., "On the Hybrid Stress Finite Element Model for Incremental Analysis of Large Deflection Problems", Int. J. of Solids & Struc., Vol. 9, 1973, pp. 1177-1191.

[34] Atluri, S.N. and Murakawa, H., "On Hybrid Finite Element Models in Nonlinear Solid Mechanics," (Eds. P. G. Bergan, et al.), Tapir Press, Norway, Vol. 1, August 1977, pp. 3-40.

[35] Murakawa, H. and Atluri, S.N., "Finite Elasticity Solutions Using Hybrid Finite Elements Based on a Complementary Energy Principle," J. of Appl. Mech., Vol. 45, 3, Sept. 1978, pp. 539-547.

[36] Murakawa, H. and Atluri, S.N., "Finite Elasticity Solutions Using Hybrid Finite Elements Based on a Complementary Energy Principle, Part II: Incompressible Materials," J. of Applied Mechanics, Vol. 46, 1, 1979, pp. 71-78.

[37] Atluri, S.N., "On Rate Principles for Finite Strain Analysis of Elastic and Inelastic Nonlinear Solids," Recent Research on Mechanical Behavior, U. of Tokyo press, 1979, pp. 79-107.

[38] Murakawa, H., and Atluri, S.N., "Finite Element Solutions of Finite-Strain Elastic-Plastic Problems, Based on a Complementary Rate Principle," Advances in Computer Methods for Partial Differential Equations, 1979, pp. 53-60.

[39] Murakawa, H., and Atluri, S.N., "Consistent Formulations of Assumed Stress Finite Element Methods Based on Consistent Rate Complementary Energy Principles for Finite Strain Problems," Proc. Int. Conf. on Computational Methods in Nonlinear Mechanics, TICOM, University of Texas at Austin, March 1979, pp. 175-177 (extended abstract only).

[40] Murakawa, H., and Atluri, S.N., "Finite-Strain Plasticity Computations Based on a new Complementary Rate Principle," 16th An. Meet. of the Society of Engineering Science, Sept. 1979, p. 20 (invited paper).

[41] Murakawa, H., and Atluri, S.N., "A Consistently Formulated Assumed Stress Finite Element Method for Finite-Strain Elasto-Plastic Analysis, Proc., Int. Conf. on Finite Element Methods in Nonlinear Problems, U. of Roorkee, India, October 1979, pp. 126-135.

[42] Atluri, S.N., "Rate Complementary Energy Principle; Finite Strain Plasticity Problems and Finite Elements," Variational Meth. in the Mechanics of Solids, (Ed.: S. Nemat-Nasser), Pergamon Press, 1980, pp. 363-367.

[43] Atluri, S.N., "On Some New General and Complementary Energy Theorems for the Rate Problems in Finite Strain, Classical Elastoplasticity," J. of Struc. Mech., Vol. 8, 1, 1980, pp. 61-92.

[44] Murakawa, H., and Atluri, S.N., "New General and Complementary Energy Theorems, Finite Strain, Rate Sensitive Inelasticity and Finite Elements : Some Computational Studies," Nonlinear Finite Element Analysis in Structural Mechanics (Eds.: W. Wunderlich, E. Stein, and K.J. Bathe), Springer-Verlag, 1981, pp. 28-48.

[45] Bratianu, C., and Atluri, S.N., "Recent Studies on Hybrid Finite Elements in Solids and Fluids," Proc., 17th Ann. Soc. of Engi. Scie. Meet., Atl., GA (extended abstract).

[46] Murakawa, H., and Atluri, S.N., "Finite deformations, Finite Rotations and Stability of Plates: A New Complementary Energy-Finite Element Analysis," Proc. 22nd AIAA/ASME/ASCE/AHS Structures, Structural Dynamics & Materials Conference, Atl., GA. April 1981, pp. 7-15.

[47] Fukuchi, M. and Atluri, S.N., "Finite Deformation Analysis of Shells: A Complementary Energy-Hybrid Approach," Nonlinear Finite Element Analysis of Plates & Shells (Ed. Hughes), AMD-Vol. 46, November 1981, pp. 233-247.

[48] Reed, K.W. and Atluri, S.N., "Visco-Plasticity and Creep: A Finite Deformation Analysis Using Stress-Based Finite Elements," Advances in Aerospace Stru. & Materi., ASME, 1981, pp. 211-219.

[49] Atluri, S.N., "Alternate Stress and Conjugate Strain Measures, and Mixed Variational Formulations Involving Rigid Rotations, for Computational Analyses of Finitely Deformed Solids, with Application to Plates and Shells - Part I: Theory," Comp. & Struc., Vol. 18, 1, 1984, pp. 93-116.

[50] Atluri, S.N., "New Developments in Numerical Analysis of Large Deformation Plasticity, Applied to Geomechanics", Proc., Int. Symp. on Numerical Models in Geomechanics, Zurich, Switzerland, 1982.

[51] Reed, K.W. and Atluri, S.N., "Inelastic Stress Analysis at Finite Deformation Through Complementary Energy Approaches," Recent Developments in Computing Methods for Nonlinear Solid & Struc. Mech., AMD, ASME, 1983, pp. 191-226.

[52] Fukuchi, N. and Atluri, S.N., "Finite Deformation Analysis of Shells: A Hybrid Finite Element Method Based on Assumed Stress-Function Vector and Rotation Tensor," Proc. 24th AIAA/ASME/ASCE/AHS Structures, Structural Dynamics & Materials Conference, Lake Tahoe, California, May 1983, pp. 205-215.

[53] Reed, K.W. and Atluri, S.N., "Hybrid Stress Finite Elements for Large Deformations of Inelastic Solids," Comp. & Struc., Vol. 19, 1-2, 1984, pp. 175-182.

[54] Rubinstein, R., Punch, E. and Atluri, S.N., "An Analysis of, and Remedies for, Kinematic Modes in Hybrid-Stress Finite Elements: Selection of Stable, Invariant Stress Fields," Comp. Meth. in Appl. Mech. & Engi., Vol. 39, 1983, pp. 245-295.

[55] Reed, K.W. and Atluri, S.N., "Analysis of Large Quasistatic Deformations of Inelastic Bodies by a New Hybrid-Stress Finite Element Algorithm," Comp. Meth. in Appl. Mech. & Engi., Vol. 39, 1983, pp. 245-295.

[56] Reed, K.W. and Atluri, S.N., "Analysis of Large Quasistatic Deformations of Inelastic Bodies by a New Hybrid-Stress Finite Element Algorithm Applications," Comp. Meth. in Appl. Mech. & Engi., Vol. 40, 1983, pp. 171-198.

[57] Atluri, S.N., "On Constitutive Relations at Finite Strain: Hypo-elasticity & Elasto-Plasticity with Isotropic or Kinematic Hardening," Comp. Meth. in Appl. Mech. & Engi., Vol. 43, 1984, pp. 137-171.

[58] Punch, E.F. and Atluri, S.N., "Applications of Isoparametric Three-Dimensional Hybrid-Stress Finite Elements with Least-Order Stress Fields," Comp. & Struc., Vol. 19, 3, 1984, pp. 409-430.

[59] Reed, K.W. and Atluri, S.N., "Inelastic Stress Analysis At Finite Deformation through Complementary Energy Approaches," Comp. Meth. for Nonlinear Solids & Struc. Mech., ASME-AMD-Vol. 54, American Society of Mech. Engi., New York, 1983, pp. 191-226.

[60] Atluri, S.N., "Mixed Finite Element Models for Plate Bending Analysis - Theory," Comp. & Struc., Vol. 19, 3, 1984, pp. 431-445.

[61] Karamanlidis, D. and Atluri, S.N., "Mixed Finite Element Models for Plate Bending Analysis: A New Element and Its Applications," Comp. & Struc., Vol. 19, 4, 1984, pp. 565-581.

[62] Reed, K.W. and Atluri, S.N., "Constitutive Modeling and Stress Analysis for Finite Deformation Inelasticity," in Constitutive Eqns. Macro and Computational Aspects, ASME, New York, 1984, pp. 111-131.

[63] Punch, E.F. and Atluri, S.N., "Least-Order, Stable, Invariant, Isoparametric Hybrid Finite Elements for Linear Continua and Finitely Deformed Plates," Proc., 25th AIAA/ASME/ASCE Struc., Struc. Dynamics & Materi. Conf., Palm Springs, California, May 1984, pp. 190-202.

[64] Punch, E.F. and Atluri, S.N., "Development and Testing of Stable, Invariant, Isoparametric Curvilinear 2- and 3-D Hybrid Stress Elements," Comp. Meth. in Appl. Mech. & Engi., Vol. 47, 1984, pp. 331-356.

[65] Xue, W-M., Karlovitz, L.A. and Atluri, S.N., "On the Existence and Stability Conditions for Mixed-Hybrid Finite Element Solutions Based on Reissner"s Variational Principle," Int. J. of Solids & Struc., Vol. 21, 1, 1985, pp. 97-116.

[66] Xue, W-M. and Atluri, S.N. "Existence and Stability, and Discrete BB and Rank Conditions, for General Mixed-Hybrid Finite Elements in Elasticity," Proc., 1985 Winter Ann. Meet., Symp. on Hybrid and Mixed Finite Element Models, Miami Beach, Florida, Nov. 17-22, 1985.

[67] Atluri, S.N., "Computational Solid Mechanics (Finite Elements and Boundary Elements): Present Status and Future Directions," Proc., 4th Inte. Conf. on Appl. Numer. Modeling, 1984, pp. 19-37.

[68] Atluri, S.N., "Notes and Comments on Computational Elasto-Plasticity: Some New Models and Their Numerical Implementation," Proc., FEICOM-85 Conf., Bombay, India, Dec. 1985, Pergamon Press.

[69] Atluri, S.N., "Constitutive Modeling of Finite Plasticity," Symp. on Advances and Trends in Computational Mechanics, ASME WAW, Anaheim, Dec. 1986.

[70] Atluri, S.N., "Analysis and Control of Finite Inelastic Deformation," Proc. 4th Int. Conf. on Innovative Numerical Methods, Atl., Ga, March 1986.

[71] Atluri, S.N., "Unification of Concepts in Small Finite Deformation Theories of Plasticity," Proc. Workshop on Constitutive Models (Army Research Office) VPI & SU, March 1986, pp. 18.

[72] Im, S., and Atluri, S.N., "Constitutive Modeling of Finite-Deformation Elasto-Plasticity, Using an Internal Time Concept," in Constitutive Laws for Engineering Materials: Theory and Applications, (C. S. Desai, et al, Eds) Elsevier Science Publ. Co., N.Y. pp. 123-136.

[73] Atluri, S.N., "An Endochronic Approach and other Topics in Small & Finite Deformation Plasticity," in Finite Elements in Nonlinear Mechanics, Springer-International, 1986, pp. 17-48.

[74] Atluri, S.N., Zhang, J.D. and O"Donoghue, P., E. "Analysis of Control of Finite Deformation of Plates and Shells," Chapter 6, in Finite Element Methods for Plates and Shell Structures (Eds. T.J.R. Hughes, and E. Hinton) Pineridge Press, Swansea, 1986, pp. 127-153.

[75] Im, S., and Atluri, S.N., "A study of Two Finite Strain Plasticity Models: An Internal Time Theory Using Mandel"s Director Concept and a Combined Isotropic-Kinematic Hardening Theory," Int. J. of Plasticity Vol. 3, pp. 163-191, 1987.

[76] Im, S., and Atluri, S.N., "Endochronic Constitutive Models of Finite Deformation Plasticity and Creep: A Field-Boundary Element Computational Algorithm," Proc. ASME WAM, Boston, Dec. 1987.

APPENDIX A

Validation Cases

Dr. Y.-T. Wu
Dr. O. H. Burnside

Southwest Research Institute

VALIDATION CASE 1

TITLE: Static Analysis of Cantilever Beam

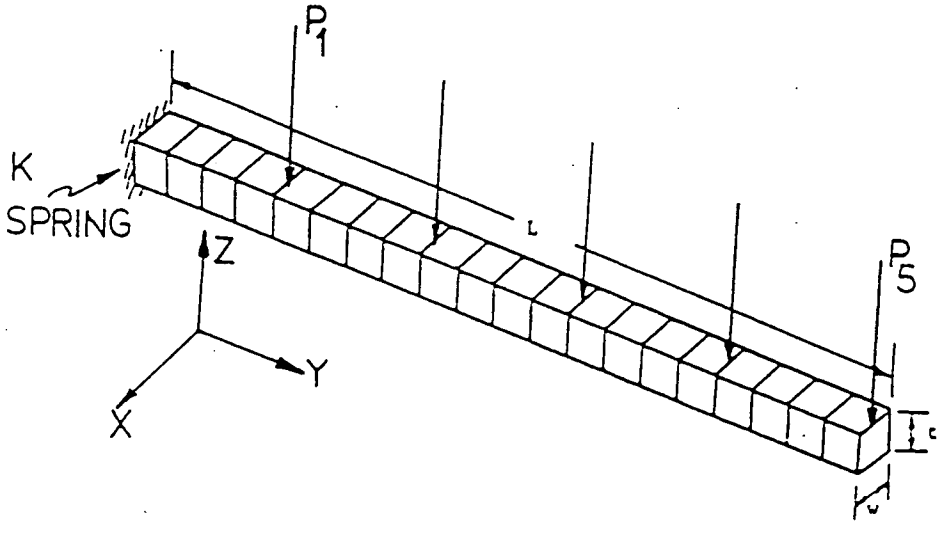
PROBLEM: A cantilever beam is subjected to correlated point loads. Determine the probabilistic distributions of the tip displacement.

TYPE: Static, correlated loading

RESPONSES: Tip displacement

FEM MODEL: NESSUS element type 98 - Two-node Timoshenko beam element
Number of elements = 20
Number of nodes = 21 (6 degrees-of-freedom per node)
Boundary conditions: Two base springs

Figure V1-1



ANALYTICAL MODEL:

Analytical Solution:

Tip displacement = Sum {2 * Pi * Li**2 * (3*L - Li) + Pi * Li * L / K}
(i = 1 to 5)

where Pi = ith load
E = Young's modulus
L = Total Length
Li = Distance from the fixed end to Pi
K = Base spring constant

Reference: PSAM 1st Annual Report, Vol. III, 1985

DEFINITION OF RANDOM/DETERMINISTIC VARIABLES

Number of Random Variables = 10

Variables	Distribution	Median	Coef. of Vari
Correlated Loads, P1 to P5*	Normal	20 lb (mean)	10%
Young's Modulus	Lognormal	10E+06 psi	3%
Length	Lognormal	20 in	5%
Thickness	Lognormal	0.98 in	5%
Width	Lognormal	1.0 in	5%
Base Spring	Lognormal	1E+05 lb-in/rad	5%

*Note: Correlation coefficients = exp(-Distance between loads/20)

NESSUS CONVERGENCE/PERTURBATION SETTINGS

1. Convergence Limit:
   - Max. number of iterations allowed: 25
   - Max. allowable rel. error in the residuals: 0.001
   - Max. allowable abs. error in the residuals: Inactive.
   - Max. allowable rel. error in the r.m.s. of displacement: Inactive
   - Max. allowable rel. error in the r.m.s of strain energy: Inactive
2. Perturbation Range:
   - +0.001 standard deviation for length.
   - +0.1 standard deviations for the remaining independent random var

SOLUTION COMPARISON:

1. Deterministic solution using mean values of random variables: (node 21, component 3)

	Tip Displacement
Theory	0.4032 in
NESSUS	0.3969 in
Difference	1.5%

2. Probabilistic solutions at selected probabilistic levels:
   Theory: Monte Carlo solution (sample size = 100,000)
   NESSUS: Mean-Value-First-Order (MVFO) solution
   Advanced MVFO solution
   First iteration solution
   (See Figure 2)

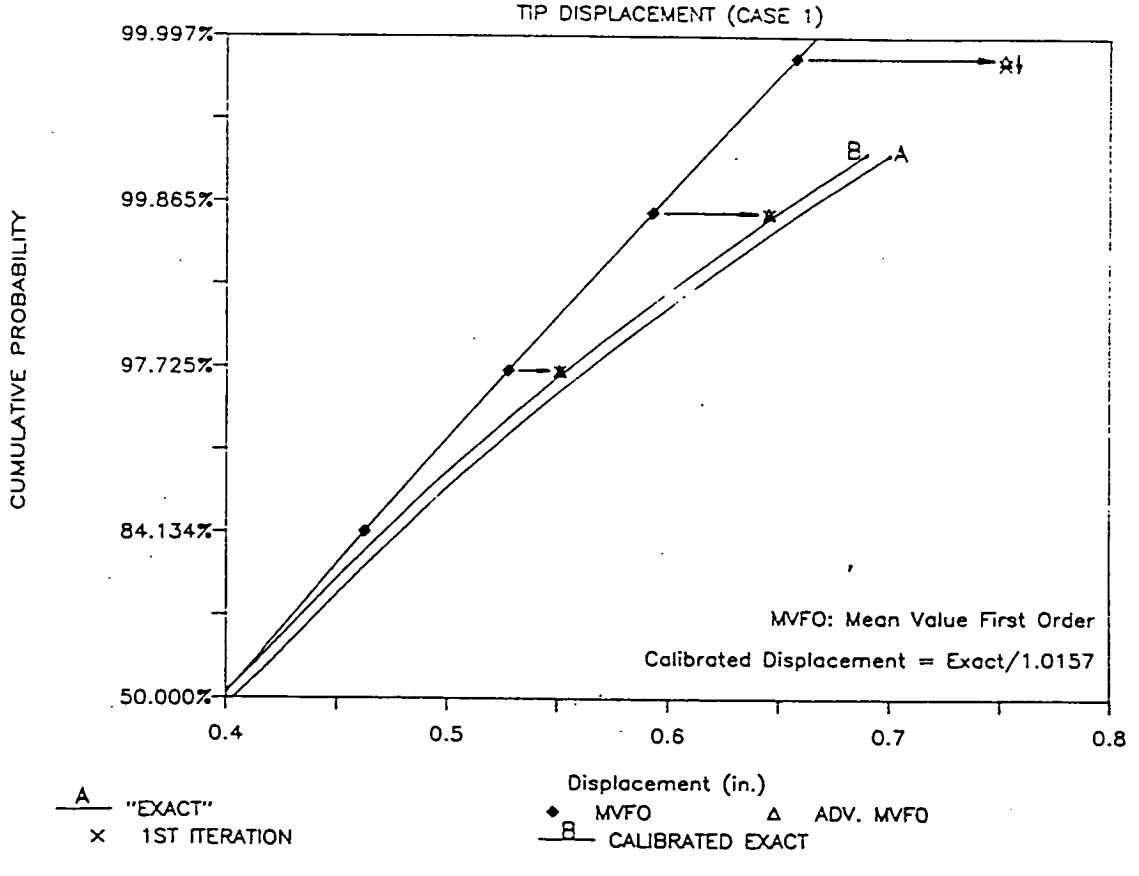
REMARKS:

1. The perturbation range for the length must be small enough, otherwise the perturbation solutions may diverge.

2. For the probabilistic solution, a calibrated 'exact' solution is derived by dividing the theoretical displacement by a factor of 1.0157. This factor is the ratio of the theoretical solution to the NESSUS solution, both computed at the mean values.

3. The output of the NESSUS code does not include stresses (moments are the standard output). The validation of the root stress is included in validation case 2 which employs plate element.

Figure V1-2



VALIDATION CASE 2

TITLE: Static Analysis of Cantilever Plate

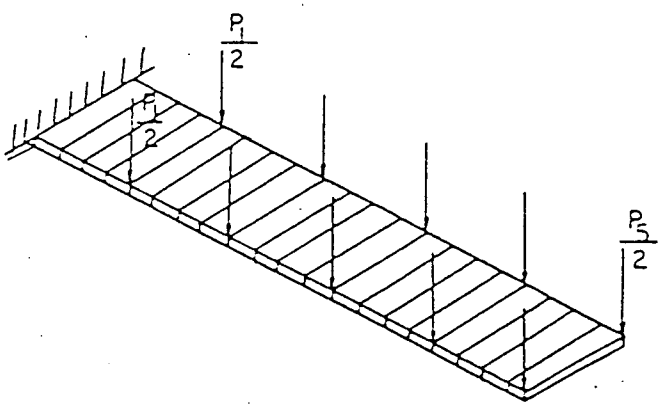
PROBLEM: A cantilever plate is subjected to correlated point loads. Determine the probabilistic distributions of the tip displacement and the root stress.

TYPE: Static, correlated loading

RESPONSES: Tip displacement and root stress

FEM MODEL: NESSUS element type 75 - Four-node shell element
Number of elements = 20
Number of nodes = 42 (6 degrees-of-freedom per node)
Boundary conditions: Two base springs

Figure V2-1



ANALYTICAL MODEL:

Analytical Solution:

Tip displacement = Sum (2 * Pi * Li**2 * (3*L - Li) + Pi * Li * L / K
(i = 1 to 5)

where Pi = ith load (loads are partially correlated)
E = Young's modulus
L = Total Length
Li = Distance from the fixed end to Pi
K = Base spring constant

Reference: FSAM 1st Annual Report, Vol. III, 1985

DEFINITION OF RANDOM/DETERMINISTIC VARIABLES

Number of Random Variables = 10

Variables	Distribution	Median	Coef. of Variation
Correlated Loads, P1 to P5*	Normal	0.1 lb (mean)	10%
Young's Modulus	Lognormal	10E+06 psi	3%
Length	Lognormal	20 in	5%
Thickness	Lognormal	0.1 in	5%
Width	Lognormal	1.0 in	5%
Base Spring	Lognormal	1E+05 lb-in/rad	5%

*Note: Correlation coefficients = exp(-Distance between loads/20)

NESSUS CONVERGENCE/PERTURBATION SETTINGS

1. Convergence Limit:
   - Max. number of iterations allowed: 8
   - Max. allowable rel. error in the residuals: 0.001
   - Max. allowable abs. error in the residuals: Inactive.
   - Max. allowable rel. error in the r.m.s. of displacement: Inactive
   - Max. allowable rel. error in the r.m.s of strain energy: Inactive
2. Perturbation Range:
   +0.01 standard deviations for the length and the width.
   +0.1 standard deviations for the remaining independent random variables.

SOLUTION COMPARISON:

1. Deterministic solution using mean values of random variables: (node 21, component 3)

	Tip Displacement	Root Stress
Theory	0.7648 in	3600 psi
NESSUS	0.7692 in	3657 psi
Difference	0.5%	1.6%

2. Probabilistic solutions at selected probabilistic levels:
   Theory: Monte Carlo solution (sample size = 100,000)
   NESSUS: Mean-Value-First-Order (MVFO) solution
   Advanced MVFO solution
   First iteration solution
   (See Figures 2 and 3 for comparison)

REMARKS:

1. The perturbation range for the length and the width must be small enough, otherwise the perturbation solutions may diverge.

2. For the probabilistic solution of stress (see Figure 3), a 'calibra exact' solution was derived by dividing the theoretical stresses by a of 0.982. This factor is the ratio of the theoretical solution to the NESSUS solution, both computed at the mean values.

Figure V2-2

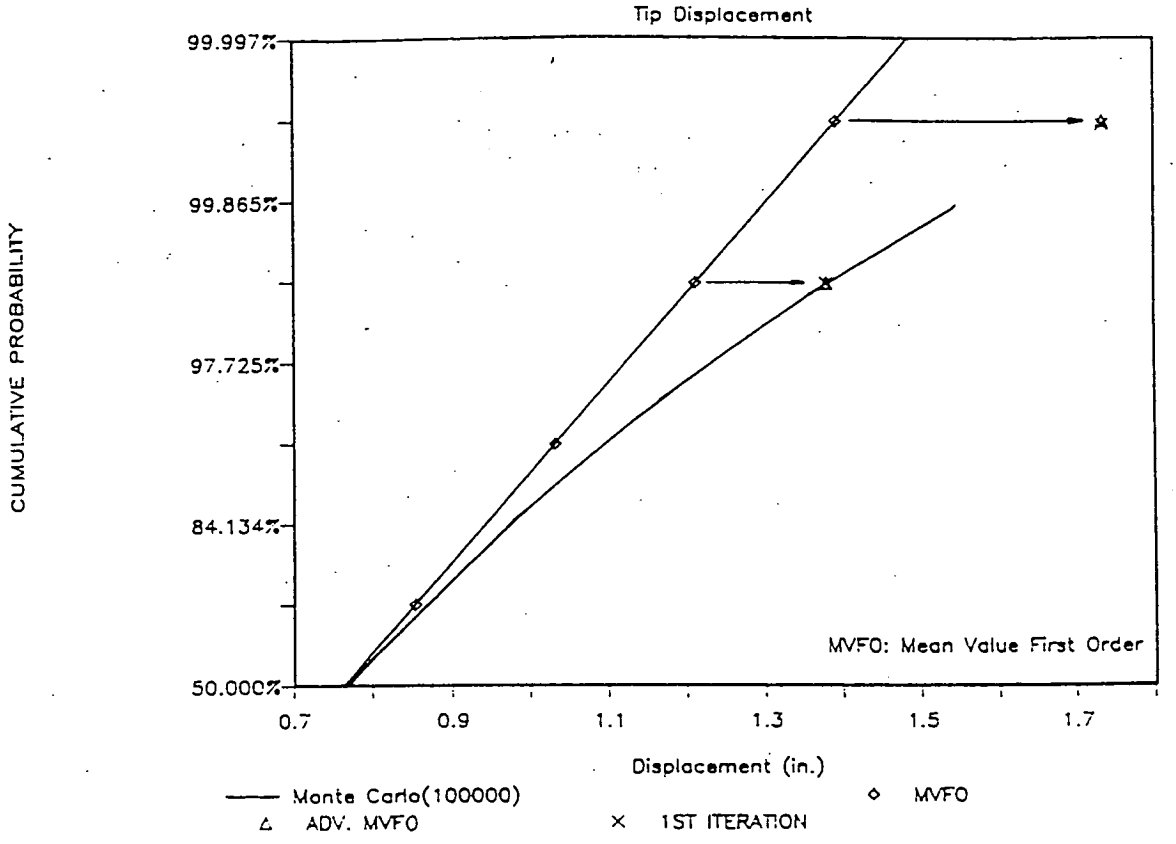
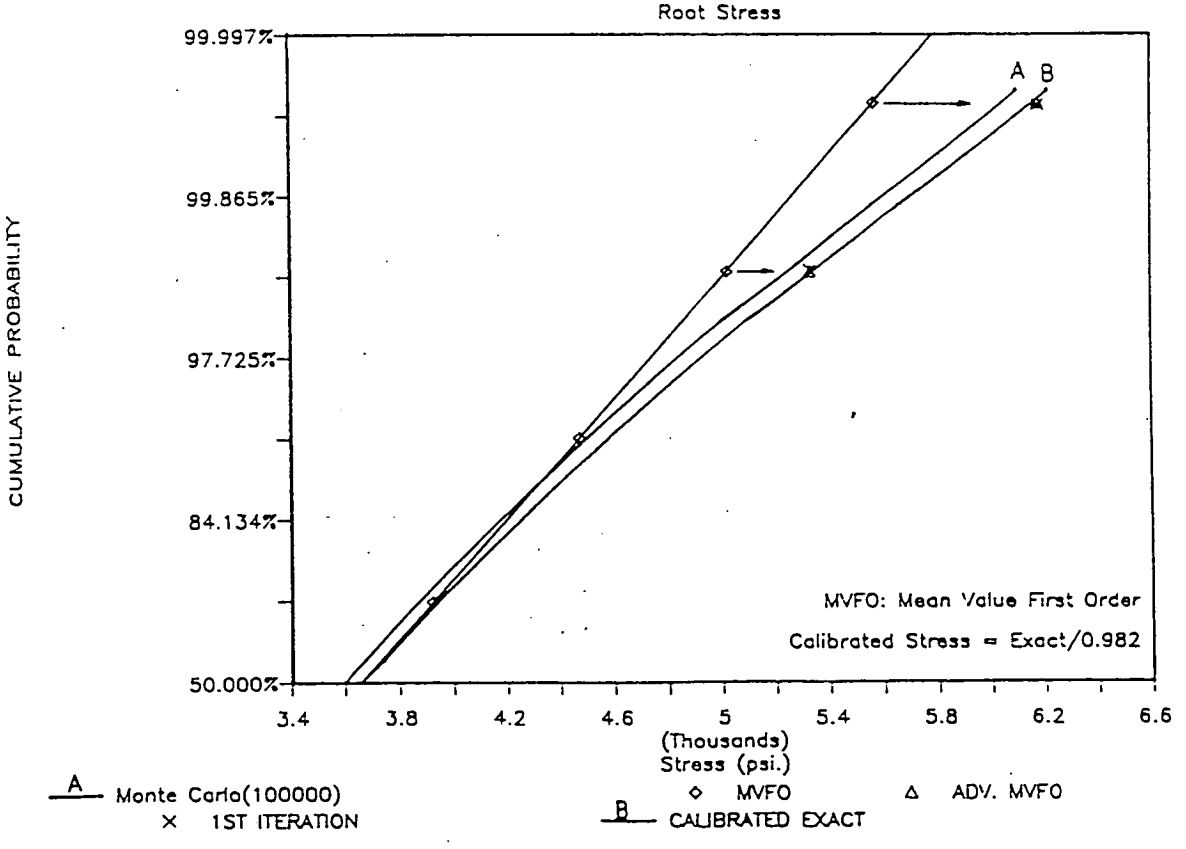


Figure V2-3



VALIDATION CASE 3

TITLE: Eigenvalue Analysis of Cantilever Beam

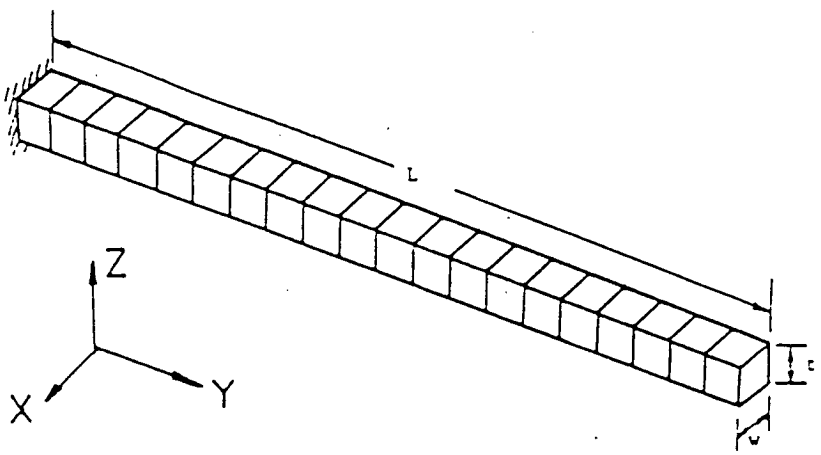
PROBLEM: Determine the probabilistic distribution of the natural frequency

TYPE: Natural Frequency

RESPONSES: First three modal frequencies in two directions

FEM MODEL: NESSUS element type 98 - Two-node Timoshenko beam element
Number of elements = 20
Number of nodes = 21 (6 degrees-of-freedom per node)
Boundary conditions: Cantilever

Figure V3-1. FEM model



ANALYTICAL SOLUTION:

Frequencies (for both Z and X directions)

$$= Ci * SQRT \{E*I/(r*w*t*L**4\}$$

where E = modulus
I = moment of inertia = w*t**3/12
r = mass density (per unit volume)
w = width
t = thickness
L = length
i = mode number
C1 = 3.52, C2 = 22.4, C3 = 61.7

Reference: Harris & Crede (Editors), Shock and Vibration Handbook, 3rd
FSAM 1st Annual Report, Vol. III, 1985

DEFINITION OF RANDOM/DETERMINISTIC VARIABLES

Number of Random Variables = 5

Variables	Distribution	Median	Coef. of Variation
Young's Modulus	Lognormal	10E+06 psi	3%
Length	Lognormal	20 in	5%
Thickness*	Lognormal	0.98 in	5%
Width*	Lognormal	1.0 in	5%
Density	Lognormal	2.5E-4 lb-sec^2/in^4	5%

*Note: See Figure V3-1.

NESSUS CONVERGENCE/PERTURBATION SETTINGS

1. Convergence Limit:
   Max. number of iterations allowed: 20
   Max. allowable rel. error: 0.01
2. Perturbation Range:
   +0.001 standard deviation for length.
   +0.1 standard deviations for the remaining random variables.

SOLUTION COMPARISON:

1. Deterministic solution using mean values of random variables:

Mode	Theory	NESSUS	% Difference	Comments
1	497.9	496.7	0.2	1st mode in Z Dir.
2	508.1	506.8	0.2	1st mode in X Dir.
3	3168.5	3099.8	2.2	2nd mode in Z Dir.
4	3233.2	3161.5	2.2	2nd mode in X Dir.
5	8727.5	8640.9	1.0	3rd mode in Z Dir.
6	8905.6	8807.6	1.1	3rd mode in X Dir.

2. Perturbation Solutions (about mean values)

Vib. Mode	Perturbed Variable X	E	L	t	w	r	Freq.;F Theory	Freq.;F NESSUS	Gradient (dF/dX) Theory	Gradient (dF/dX) NESSUS	Percent Diff.
1	Mean	1.000E+07	20.000	0.9800	1.000	2.500E-04	497.9	496.7			
1	E	1.003E+07					498.7	497.4	2.5E-05	2.5E-05	0.3
1	L		20.001				497.9	496.6	-5.0E+01	-4.8E+01	2.9
1	t			0.9849			500.4	499.1	5.1E+02	5.1E+02	0.3
1	w				1.005		497.9	496.7	0.0E+00	0.0E+00	0.0
1	r					2.513E-04	496.7	495.4	-9.9E+05	-9.9E+05	0.0
2	Mean	1.000E+07	20.000	0.9800	1.000	2.500E-04	508.1	506.8			
2	E	1.003E+07					508.8	507.5	2.5E-05	2.5E-05	0.2
2	L		20.001				508.0	506.7	-5.1E+01	-5.0E+01	0.9
2	t			0.9849			508.1	506.8	0.0E+00	0.0E+00	0.0
2	w				1.005		510.6	509.3	5.1E+02	5.1E+02	0.4
2	r					2.513E-04	506.8	505.5	-1.0E+06	-1.0E+06	-0.0
3	Mean	1.000E+07	20.000	0.9800	1.000	2.500E-04	3168.5	3099.8			
3	E	1.003E+07					3173.2	3104.4	1.6E-04	1.5E-04	2.2
3	L		20.001				3168.2	3099.5	-3.2E+02	-3.0E+02	3.8
3	t			0.9849			3184.3	3115.0	3.2E+03	3.1E+03	4.2
3	w				1.005		3168.5	3099.8	0.0E+00	0.0E+00	0.0
3	r					2.513E-04	3160.6	3092.0	-6.3E+06	-6.2E+06	1.9
4	Mean	1.000E+07	20.000	0.9800	1.000	2.500E-04	3233.2	3161.5			
4	E	1.003E+07					3238.0	3166.3	1.6E-04	1.6E-04	2.3
4	L		20.001				3232.8	3161.2	-3.2E+02	-3.1E+02	3.6
4	t			0.9849			3233.2	3161.5	0.0E+00	0.0E+00	0.0
4	w				1.005		3249.3	3177.0	3.2E+03	3.1E+03	4.3
4	r					2.513E-04	3225.1	3153.6	-6.4E+06	-6.3E+06	2.0
5	Mean	1.000E+07	20.000	0.9800	1.000	2.500E-04	8727.5	8640.9			
5	E	1.003E+07					8740.6	8653.9	4.4E-04	4.3E-04	1.0
5	L		20.001				8726.6	8640.1	-8.7E+02	-8.4E+02	3.9
5	t			0.9849			8771.2	8681.9	8.9E+03	8.4E+03	6.0
5	w				1.005		8727.5	8640.9	0.0E+00	0.0E+00	0.0
5	r					2.513E-04	8705.8	8619.3	-1.7E+07	-1.7E+07	0.7
6	Mean	1.000E+07	20.000	0.9800	1.000	2.500E-04	8905.6	8807.6			
6	E	1.003E+07					8919.0	8820.8	4.4E-04	4.4E-04	1.1
6	L		20.001				8904.7	8806.7	-8.9E+02	-8.5E+02	4.4
6	t			0.9849			8905.6	8807.6	0.0E+00	0.0E+00	0.0
6	w				1.005		8950.2	8849.3	8.9E+03	8.3E+03	6.3
6	r					2.513E-04	8883.4	8785.6	-1.8E+07	-1.8E+07	0.9

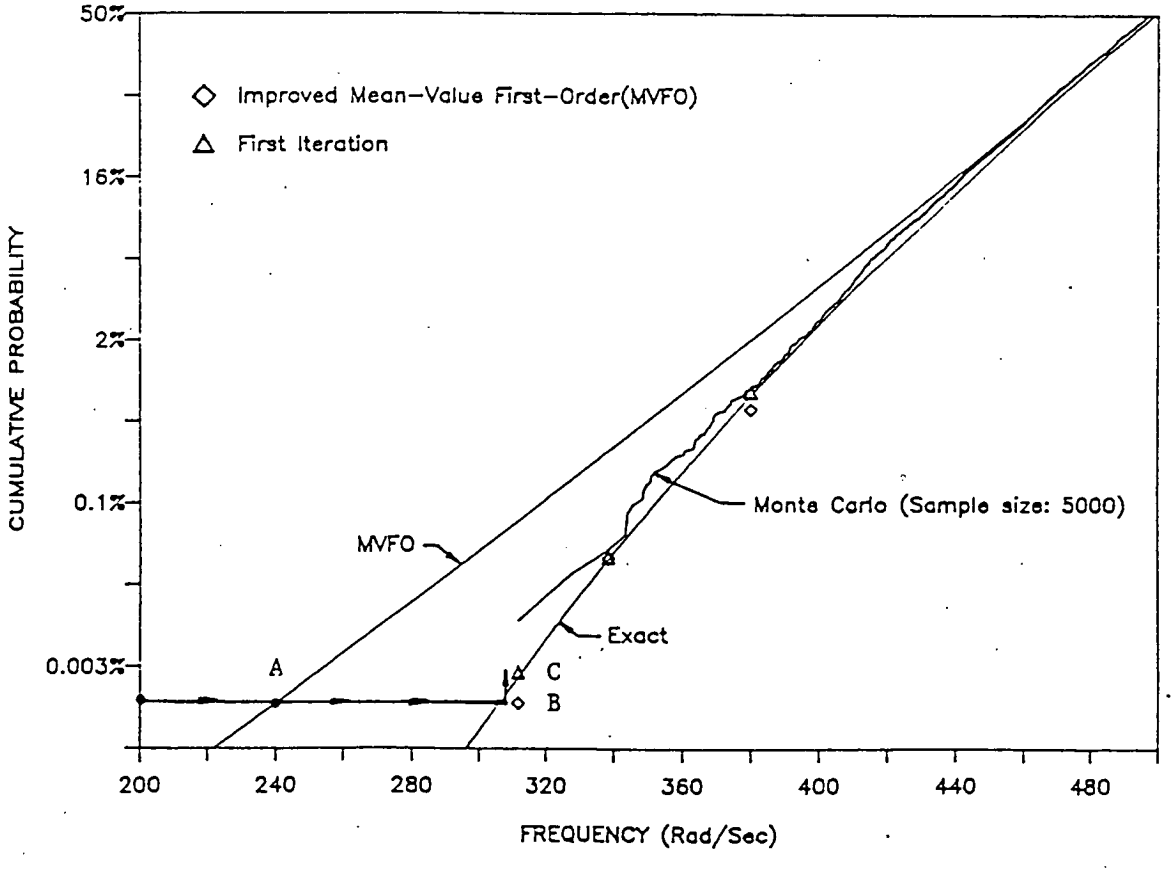
3. Probabilistic solutions for the first mode frequency at selected probabilistic levels:
   Theory: Exact CDF based on analytical solution
   Simulation: Monte Carlo (sample size = 5,000)
   NESSUS: Mean-Value-First-Order (MVFO) solution
   Advanced MVFO solution
   First iteration solution
   (See Figure V3-2)

REMARKS:

1. The perturbation range for the width must be very small, otherwise the perturbation solutions may diverge.

2. The median width (1.0 in) and thickness (0.98 in) were deliberately chosen to be slightly different to validate the NESSUS's capability to identify near roots in eigenvalue analysis.

Figure V3-2 CDF of First Mode Natural Frequency



VALIDATION CASE 5

TITLE: Rotating Beam (plate elements)

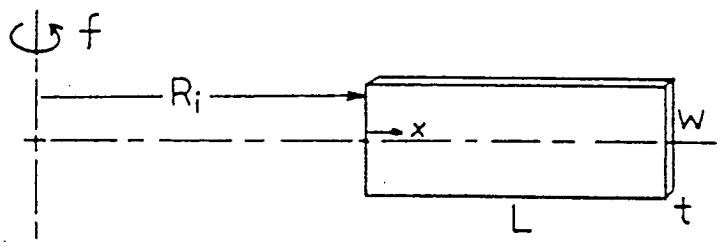
PROBLEM: Determine the probabilistic distributions of the first bending natural frequency and the tip displacement of a rotating beam

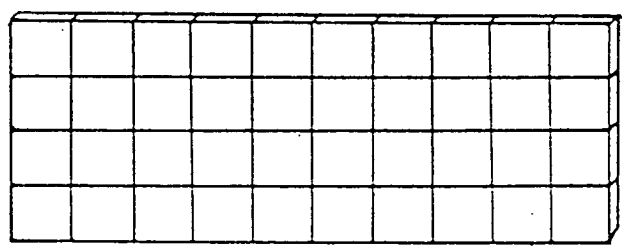
TYPE: Centrifugal loading and stress stiffening effects

RESPONSES: First bending frequency and tip displacement

FEM MODEL: NESSUS element type 75 - Four-node shell element
Number of elements = 40
Number of nodes = 55 (6 degrees-of-freedom per node)
Boundary condition: cantilevered

Figure V5-1. Sketch and FEM model





ANALYTICAL SOLUTION:

Assumed first bending mode shape: (x**4 -4*L*x**3 + 6*L**2*x**2)/L**4

Frequency = SQRT { 1.0384 * E * t**2 / (r*L**4) + (1.173+6.6/L) * f**

Tip displacement = r * (f**2) * (l**3) * (1 + Ri/L) / (3.*E)

where
E = modulus
r = mass density
w = width
t = thickness
L = length
f = rotating frequency = 400 rad/sec
Ri= inside radius = 4.237 in.

DEFINITION OF RANDOM/DETERMINISTIC VARIABLES

Number of Random Variables = 5

Variables	Distribution	Median	Coef. of Variation
Young's Modulus	Lognormal	29E+06 psi	10%
Length	Lognormal	3.844 in	5%
Thickness	Lognormal	0.0416 in	5%
Width	Lognormal	1.424 in	5%
Density	Lognormal	9E-4 lb-sec^2/in^4	5%
Rotating Frequency	Fixed	400 rad/sec	
Radius Ri*	Fixed	4.237	

*Note: see Figure V5-1

NESSUS CONVERGENCE/PERTURBATION SETTINGS (NESSUS 2.5)

1. Modal extraction:
   *MODAL 3 0 1
2. Convergence criteria:
   Increment 0:
   *ITER 0 5
   20 1.E-04
   Increment 1:
   *ITER 0 5
   20 1.E-06
3. Perturbation Settings:
   +0.001 standard deviation for length.
   +0.1 standard deviations for the remaining random variables.

SOLUTION COMPARISON:

1. Deterministic solutions using the mean values of random variables:

Table V5-1 Comparisons of the deterministic solutions

	Theory	NESSUS	NESSUS/Theory
Frequency	853.0	862.4	1.01
Tip displacement	2.4945E-4	2.4797E-4	0.994

2. Probabilistic solutions for the frequency and the displacement at selected probabilistic levels:
   Simulation: Monte Carlo (sample size = 500,000)
   NESSUS: Mean-Value-First-Order (MVFO) solution
   Advanced MVFO solution
   First iteration solution
   (See Figures V5-2 and V5-3)

REMARKS: Date: 10/16/87 NESSUS 2.5

1. The selection of the perturbation range for the length is very cri as illustrated in the following table:

Perturbation range	Results
+1.0 std.	No solution (instability)
+0.1 std.	Incorrect solution (frequency decreas
+0.001 std.	Correct solution

   The NESSUS eigenvalue perturbation algorithm needs to be reviewed.

2. The 'adjusted' exact curves in Figures V5-2 and V5-3 are defined u the ratios of the NESSUS mean solutions to the theoretical mean solutions. (see Table V5-1)

Figure V5-2 First Bending Frequency

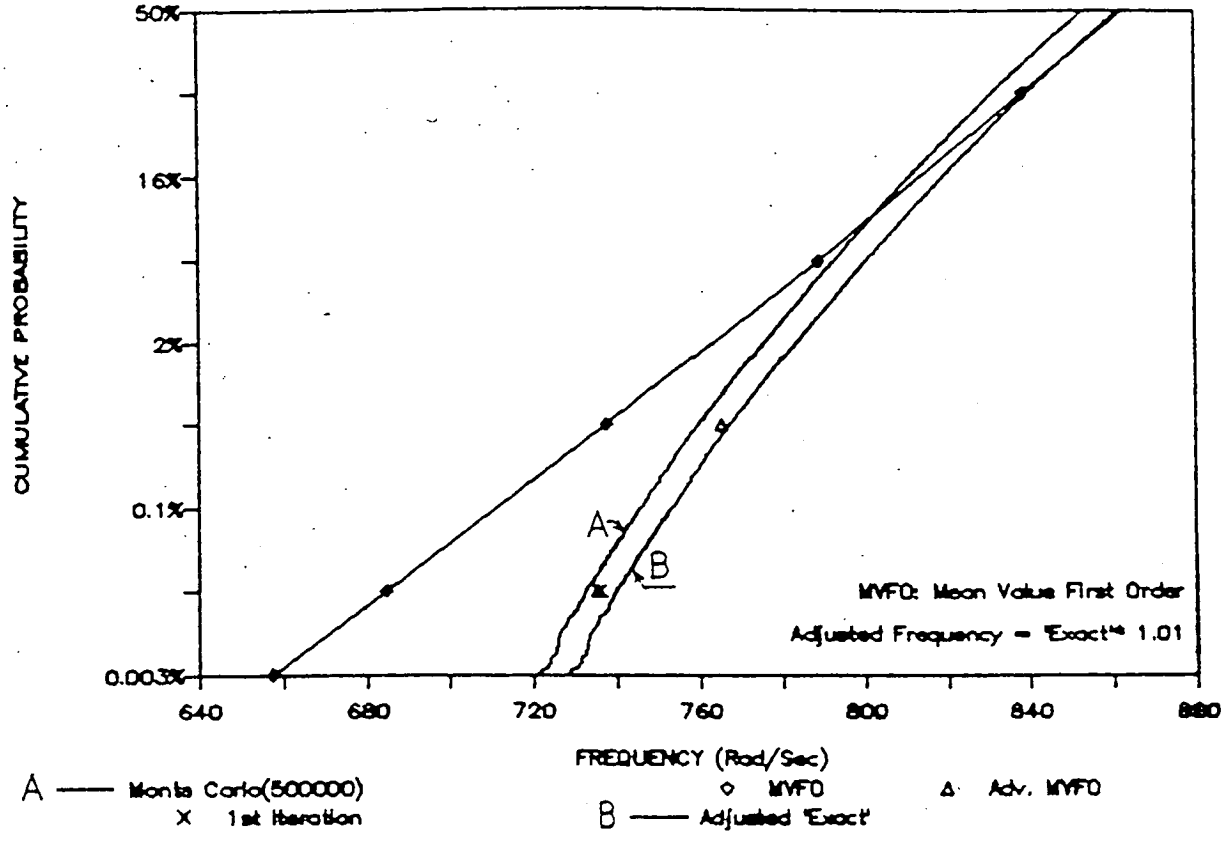
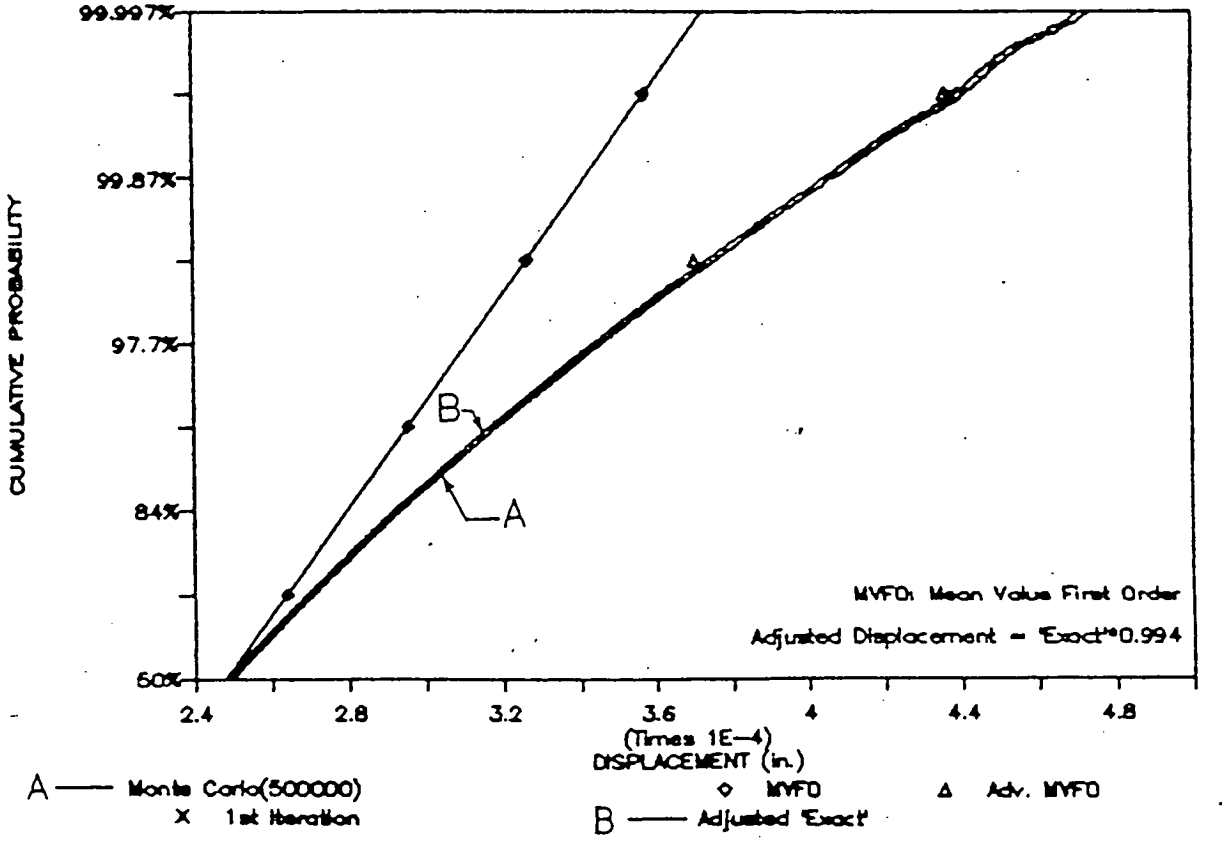


Figure V5-3 Tip Displacement



VALIDATION CASE 6

TITLE: Eigenvalue Analysis of Twisted Cantilever Plate

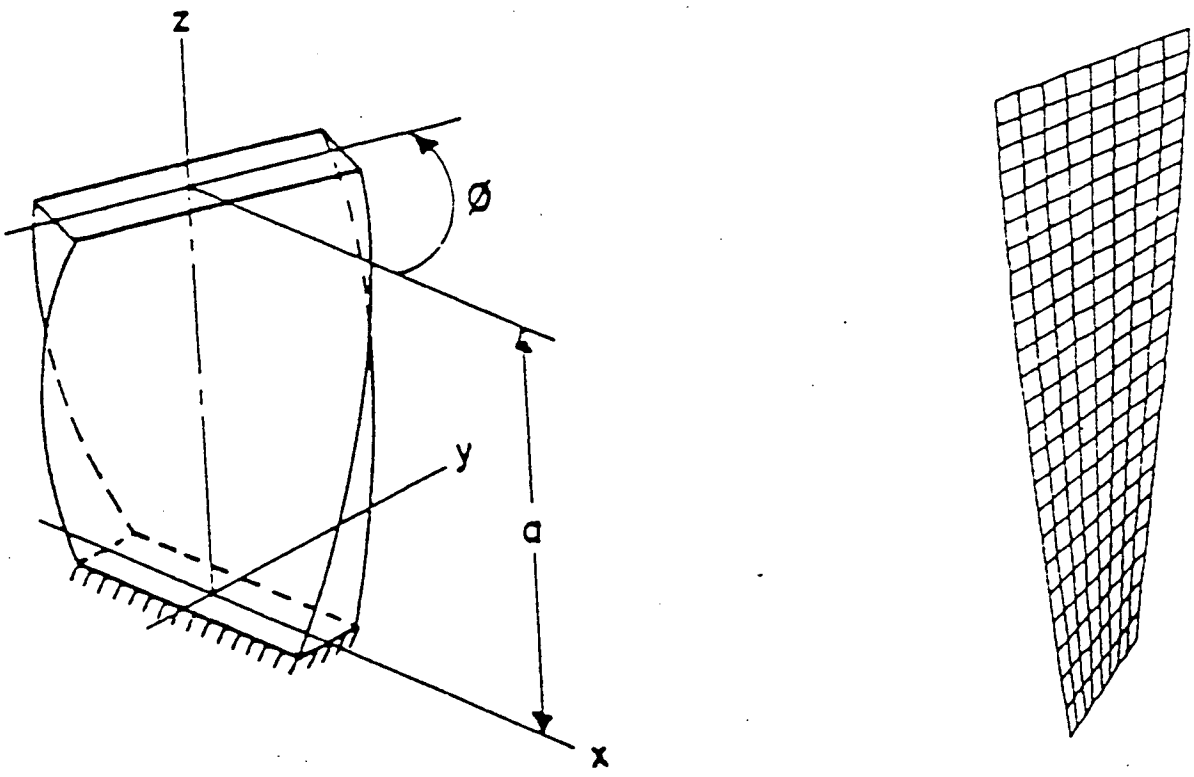
PROBLEM: Determine the probabilistic distribution of the natural frequencies

TYPE: Natural Frequency

RESPONSES: First bending and torsional modal frequencies

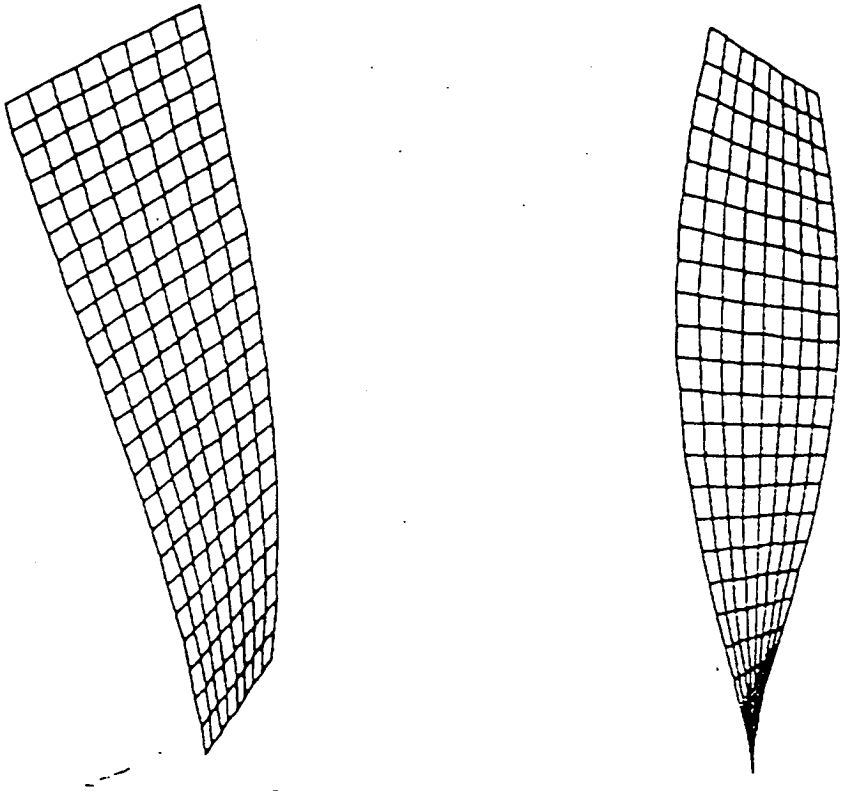
FEM MODEL: NESSUS element type 75 - Four-node shell element
Number of elements = 192
Number of nodes = 225 (6 degrees-of-freedom per node)
Boundary conditions: Cantilever

Figure 1. Sketch and FEM model



VALIDATION CASE 6 (Continued)

Figure 2. First bending and torsion mode



ANALYTICAL SOLUTION:

First bending frequency (use flat plate solution)

$$= 3.31 * SQRT \{E*h**2/(12*L**4*r*(1-v**2)\}$$

where E = modulus
r = mass density (per unit volume)
h = thickness
L = length
v = Poisson's ratio

First torsional frequency: not available

Experimental results: see Reference

Reference: Macbain, J. C.; Kielb, R. E. & Leissa, A. W., "Vibrations Twisted Cantilevered Plates - Experimental Investigation" 29th International Gas Turbine Conference, Amsterdam, The Netherland, 1984. ASME paper 84-GT-96

DEFINITION OF RANDOM/DETERMINISTIC VARIABLES

Number of Random Variables = 3

Variables	Distribution	Median	Coef. of Vari.
Young's Modulus	Lognormal	10.34E+06 psi	3%
Thickness	Lognormal	0.1 in	5%
Density	Lognormal	2.61E-4 lb-sec^2/in^4	5%
Twisted Angle	Deterministic	45 degrees	-
Length	Deterministic	6 in	-
Width	Deterministic	2.0 in	-
Poisson's ratio	Deterministic	0.3	-

NESSUS CONVERGENCE/PERTURBATION SETTINGS

1. Convergence Limit:
   Max. number of iterations allowed: 30
   Max. allowable rel. error: 0.001
2. Perturbation Range:
   +0.1 standard deviations for all the random variables.

Figure 3

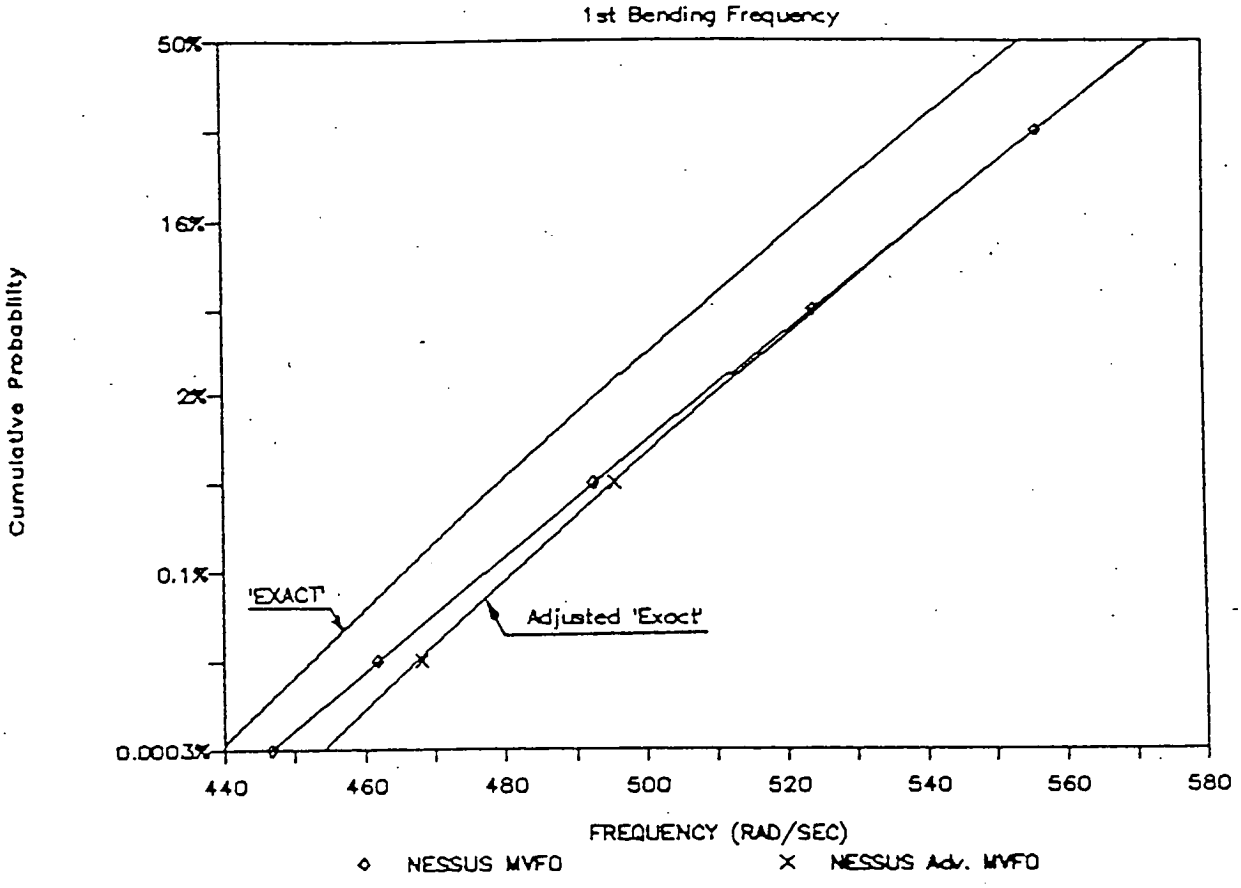
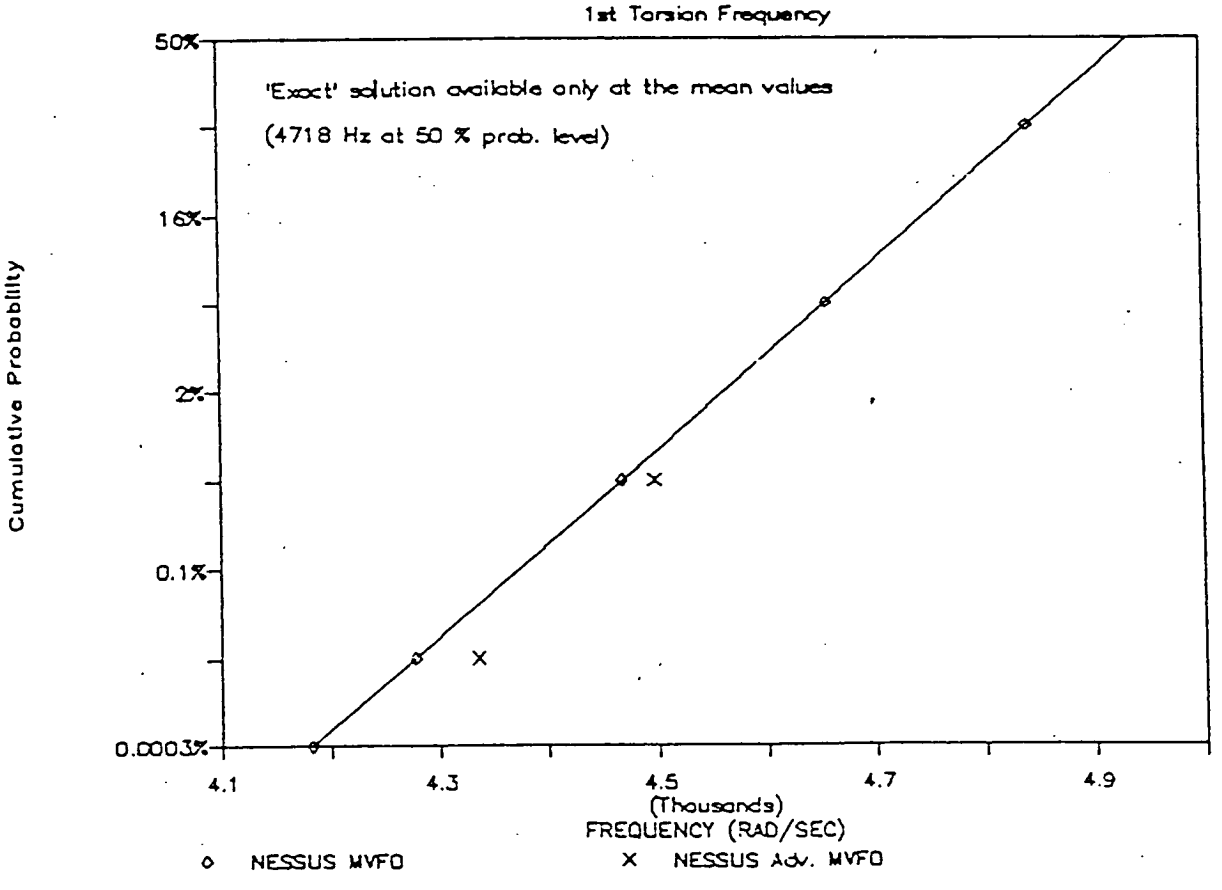


Figure 4



SOLUTION COMPARISON:

1. Deterministic solution using mean values of random variables:

Mode	Experiment	NESSUS	% Difference
Bending	553.8	572.4	3.3
Torsion	4718.2	4933.1	4.5

2. Probabilistic solutions for the first bending frequency at selecte probabilistic levels:

   Theory: Exact CDF based on analytical solution
   NESSUS: Mean-Value-First-Order (MVFO) solution
   Advanced MVFO solution
   (See Figure 3 for comparison)

   Probabilistic solutions for the first torsior frequency at selecte probabilistic levels:

   Exact: Only 50 % probability level experiment result availal
   NESSUS: Mean-Value-First-Order (MVFO) solution
   Advanced MVFO solution
   (See Figure 4)

REMARKS:

1. The analytical solution for the first bending mode was based on th flat plate solution, therefore should be considered as approximate so only. However, based on experimental investigation (see Ref.), the analytical solution predicts well for different thickness.
2. For the first bending mode, a calibrated (or adjusted) 'exact' probabilistic solution was derived by multiplying the experimental re by a factor of 1.033. This factor is the ratio of the FEM solution, at the mean values to the experimental result.
3. For the first torsional mode, the analytical solution for the flat can not be used as an approximation because the experimental results do not follow the analytical solution.

VALIDATION CASE 7

TITLE: Static Analysis of Simply Supported Plate

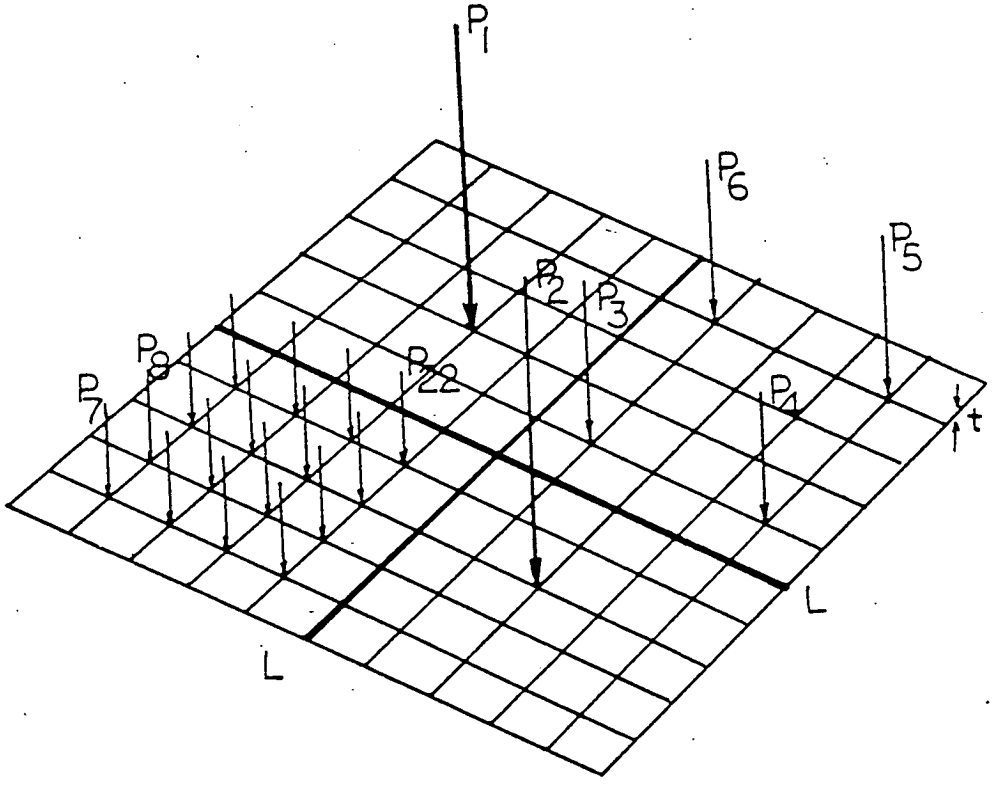
PROBLEM: A simply supported rectangular plate is subjected to point loads. Determine the probabilistic distribution of the maximum (center) displacement.

TYPE: Static, correlated loading (Multiple zones)

RESPONSES: Maximum displacement

FEM MODEL: NESSUS element type 75 - Four-node shell element
Number of elements= 100
Number of nodes= 121 (6 degrees-of-freedom per node)
Boundary conditions: Simply supported

Figure 1. FEM Model



ANALYTICAL SOLUTION:

Max. displacement = 48 * (1-v**2) / (pi**4 * E * t**3 *L**2)

[1574(P1+P2) + 2373*P3 + 676.6*P4 + 207.7*P6 + 16917*P7]

where E = modulus of elasticity
v = Poisson's ratio
t = thickness
L = Length

Reference: Timoshenko and Woinowsky-Krieger, Theory of Plates and Shells, 2nd ed., p111

DEFINITION OF RANDOM/DETERMINISTIC VARIABLES

Number of Random Variables (a) = 25

Variables	Distribution	Mean	Coef. of Var
Correlated Loads,			
P1(c)	EVD(b)	15 lb	10%
P2(c)	EVD	15 lb	10%
P3 to P6 (d)	Normal	10 lb	10%
P7 to P22 (e)	Lognormal	2 lb	10%
Young's Modulus	Weibull	10.5E+06 psi	3%
Poisson's ratio	Lognormal	0.25	3%
Thickness	Lognormal	0.1 in	5%
Width	Deterministic	10 in	-

Notes:
(a) Number of independent random variables = 10
(b) Type I extreme value distribution
(c) Independent
(d) Partially correlated with
correlation coefficients = exp{-Distance between loads/9}
(e) Fully correlated

NESSUS CONVERGENCE/PERTURBATION SETTINGS

1. Convergence Limit:
   Max. number of iterations allowed: 100
   Max. allowable rel. error in the residuals: 0.015
   Max. allowable abs. error in the residuals: 15.0
   Max. allowable rel. error in the r.m.s. of displacement: 0.002
   Max. allowable rel. error in the r.m.s of strain energy: 0.002
2. Perturbation Range:
   +0.1 standard deviations for all the independent random variables

SOLUTION COMPARISON:

1. Deterministic solution using mean values of random variables:
   (node 61; component 3)

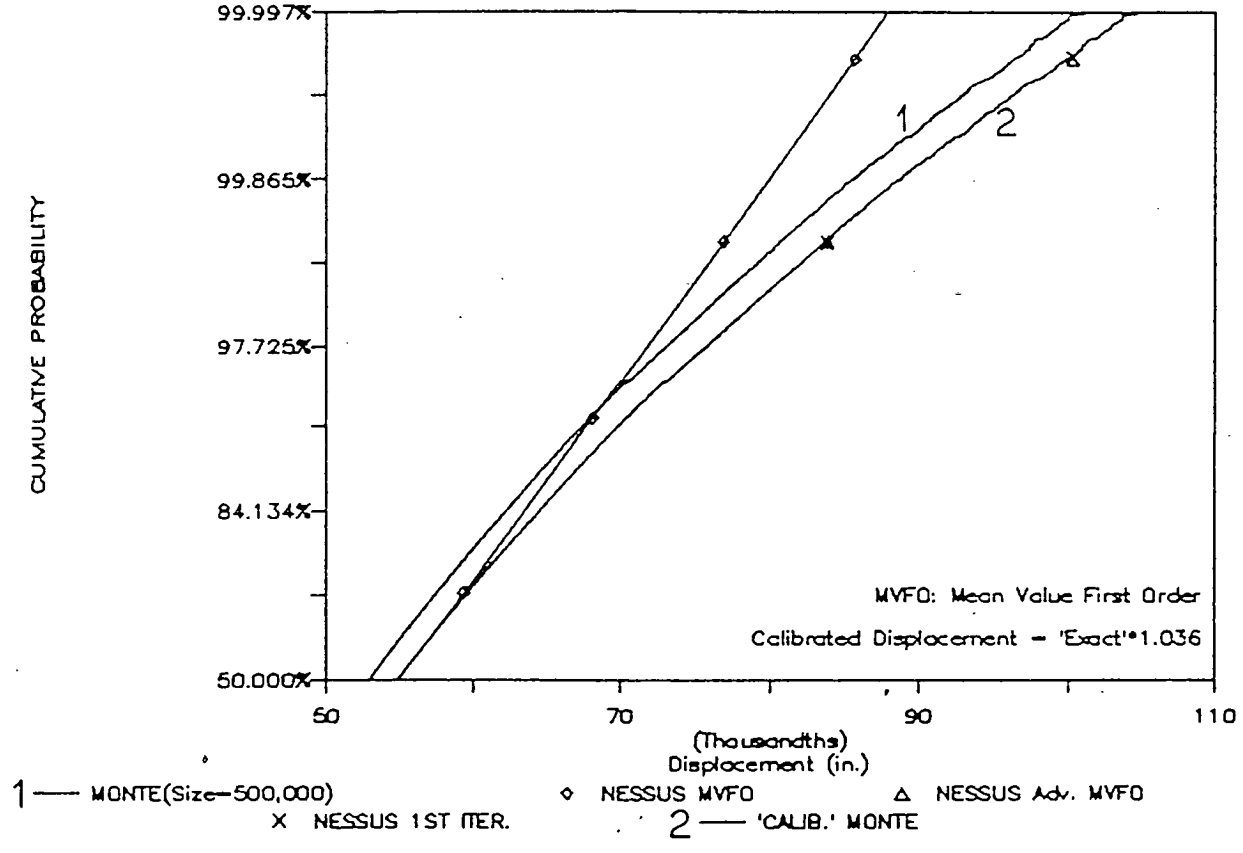
	Tip Displacement
Theory	0.05297 in
NESSUS	0.05493 in
Difference	3.7%

VALIDATION CASE 7 (Continued)

2. Probabilistic solutions at selected probabilistic levels:
   Exact: Monte Carlo simulation (sample size= 500,000)
   based on analytical solution
   NESSUS: Mean-Value-First-Order (MVFO) solution
   Advanced MVFO solution
   First iteration solution
   (See Figures 2 for comparison)

REMARKS:
1. For the probabilistic solution of displacement (Figure 2), a 'calibrated' or adjusted 'exact' solution is derived by multiplying the theoretical displacement by a factor of 1.036. This factor is the ratio of the theoretical solution to the NESSUS solution, both computed at the mean values of the random variables.

Figure 2.



VALIDATION CASE 9

TITLE: Static Analysis of Cylindrical Shell

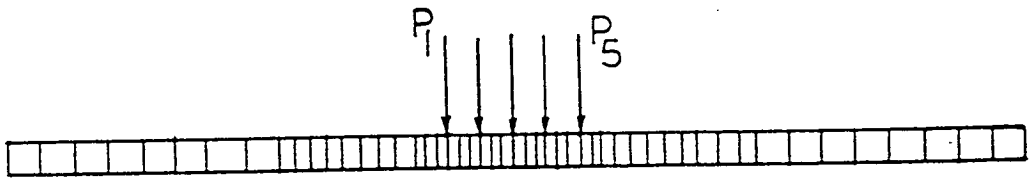
PROBLEM: A cylindrical shell is subjected to correlated point loads Determine the probabilistic distribution of the maximum displacement.

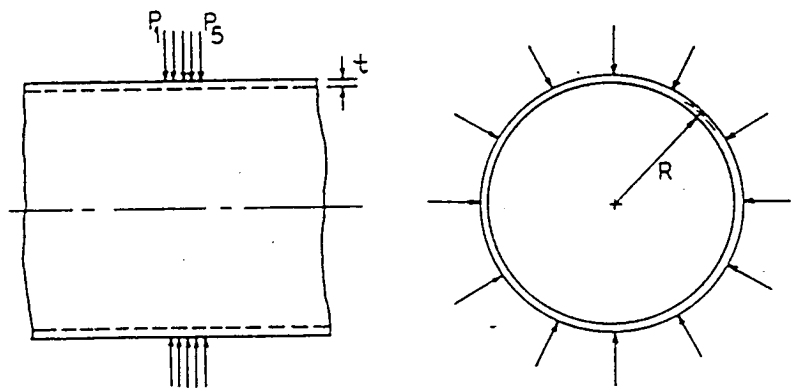
TYPE: Static, correlated loading

RESPONSES: Displacement

FEM MODEL: NESSUS element type 153 Four-node assumed strain axisymmet
Number of elements = 50
Number of nodes = 102 (2 degrees-of-freedom per node)
Boundary condition : constrined z-direction displacement a
26 and 77

Figure V9 - 1





ANALYTICAL MODEL:

Analytical Solution: See Reference
Reference: Timoshenko and Woinowsky-Krieger, Theory of Plates and She
2nd ed., p111

DEFINITION OF RANDOM/DETERMINISTIC VARIABLES

Number of Random Variables = 8

Variables	Distribution	Mean	Coef. of Variation
Correlated Loads,			
F1 to F5*	Normal	1000 lb	10%
Young's Modulus	Lognormal	29E+06 psi	3%
Thickness	Lognormal	0.1 in	5%
Mean Radius	Lognormal	2.5 in	5%
Poisson's ratio	Deterministic	0.3	0%

*Note: Correlation coefficients = exp{-Distance between loads/0.2}

NESSUS CONVERGENCE/PERTURBATION SETTINGS

1. Convergence Limit:

Max. number of iterations allowed:	120
Max. allowable rel. error in the residuals:	0.02
Max. allowable abs. error in the residuals:	20
Max. allowable rel. error in the r.m.s. of displacement:	0.01
Max. allowable rel. error in the r.m.s of strain energy:	0.055

2. Perturbation Range:
   +0.1 standard deviations for all the independent random variables.

SOLUTION COMPARISON:

1. Deterministic solution using mean values of random variables:

	Displacement
Theory	0.00797 in.
NESSUS	0.008145 in.
Difference	2.2 %

2. Probabilistic solutions at selected probabilistic levels:
   Theory: Monte Carlo solution (sample size = 500,000)
   NESSUS: Mean-Value-First-Order (MVFO) solution
   Advanced MVFO solution
   First iteration solution
   (See Figure V9-2)

REMARKS:

1. The perturbation range was chosen as 0.1 standard deviation for each random variables. It was found that NESSUS/FEM solution required very tight convergence limits for generating accurate Young's modulus sensitivity data. Also, it was found that this convergence problem ca be solved by increasing the perturbation range to 0.5 standard deviation.

2. For the probabilistic solution (see Figure V9-2), a calibrated 'exa solution was derived by multiplying the theoretical solution by a fact of 1.02. This factor is the ratio of the NESSUS solution to the theoretical solution, both computed at the mean values.

3. Because thickness is not a standard input, it is necessary to provi thickness information in terms of the coordinates (i.e., inside and ou radius). Also, the perturbation solution for the thickness must be obt by perturbing simultaneously the inside and the outside radius.

Figure V9-2 NESSUS Validation

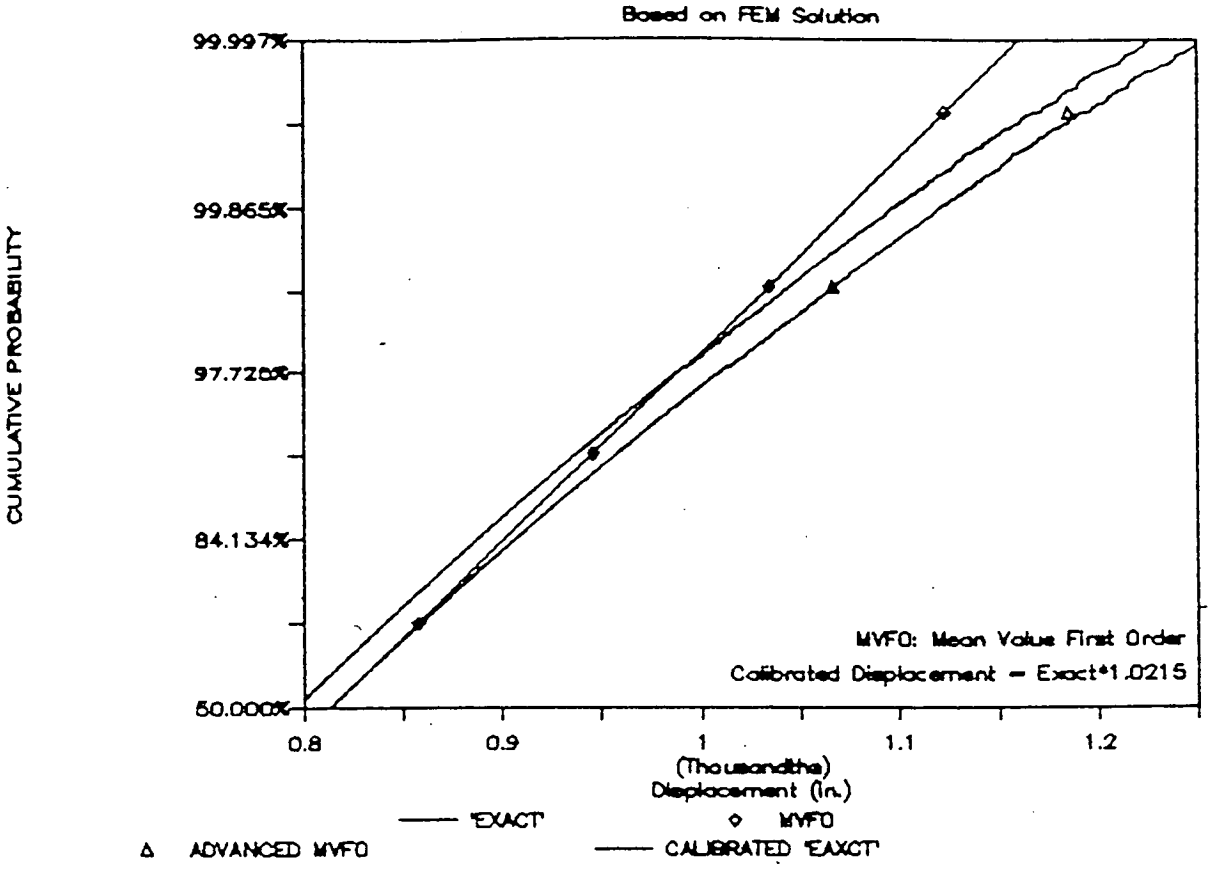
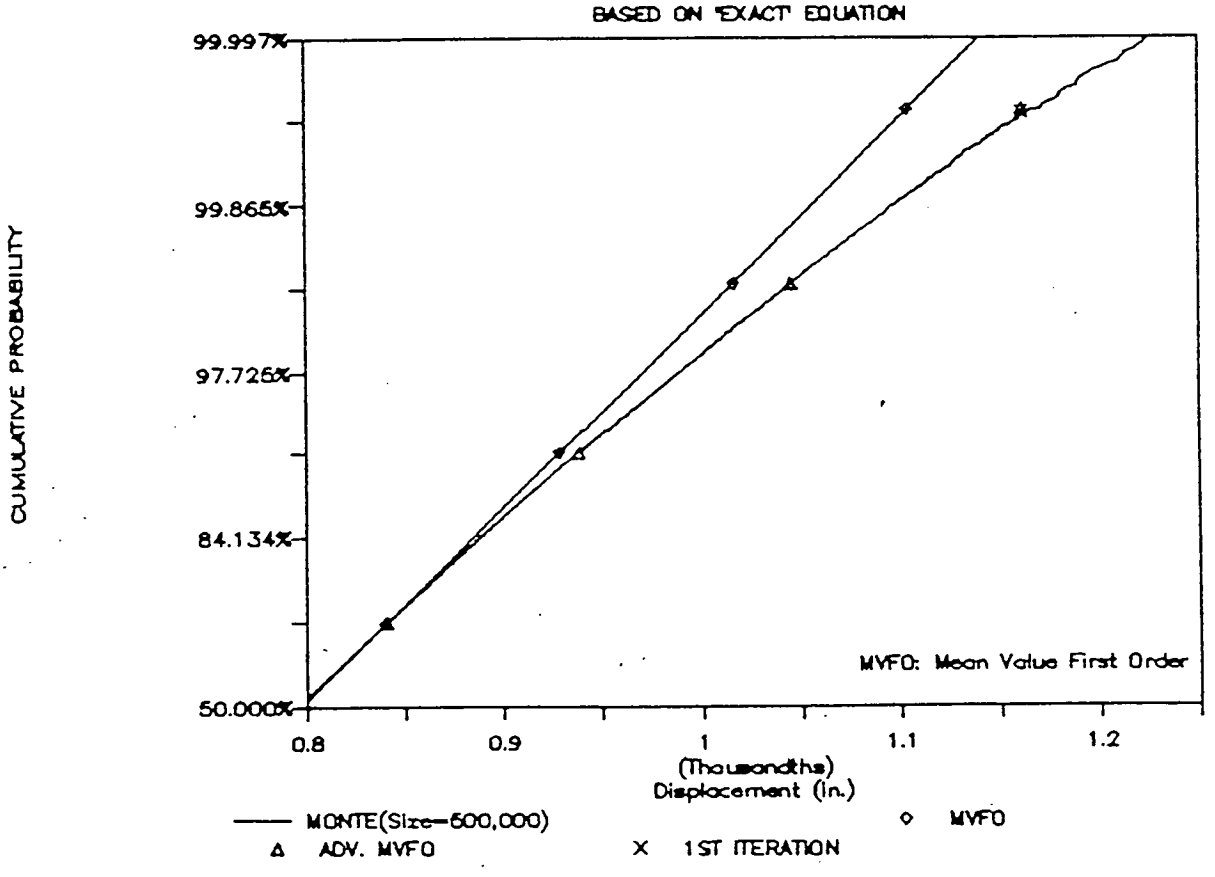


Figure V9-3 FPI Validation



VALIDATION CASE 10

TITLE: Stress Concentration Analysis

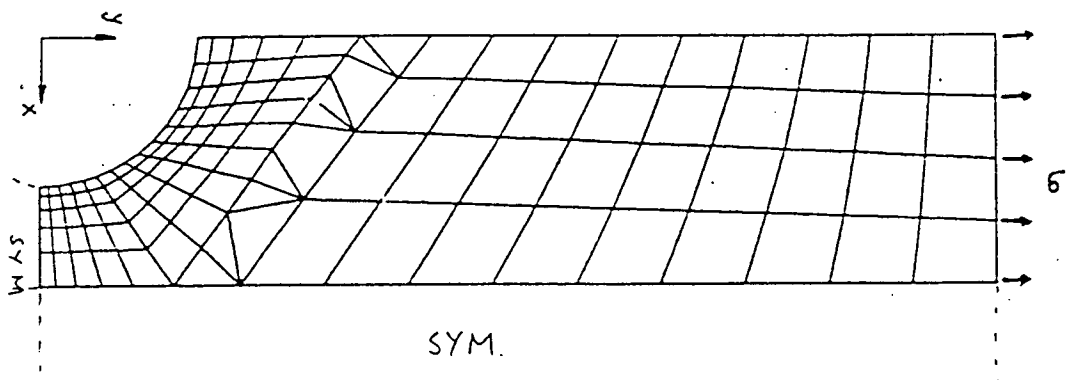
PROBLEM: Two U notches in a member of rectangular section. Determi the probabilistic distribution of the maximum stress.

TYPE: Static loading

RESPONSES: Maximum stress

FEM MODEL: NESSUS element type 3 - Four-node plane stress element
Number of elements = 117
Number of nodes = 140 (2 degrees-of-freedom per node)
Symmetry conditions along longitudinal axis of the member
Symmetry conditions across the center of the member
(one quarter of the member modeled)
Constant tensile stress applied at the y = max. boundary

Figure V10-1



ANALYTICAL MODEL:

Analytical Solution: See Reference

Reference: R. J. Roark and W. C. Young, Formulas for Stress and Stra page 590

DEFINITION OF RANDOM/DETERMINISTIC VARIABLES

Number of Random Variables = 1

Variables	Distribution	Mean	Coef. of Variation
Radius	Case A. Lognormal	2.4	2 %
	Case B. Truncated Normal*	2.4	2 %
Load	Deterministic	8000 lb	
thickness	Deterministic	0.1 in	

*Note. Truncated at +/- 3 standard deviation (2.25 to 2.55)

NESSUS CONVERGENCE/PERTURBATION SETTINGS

1. Convergence Limit:
   Max. number of iterations allowed: 30
   Max. allowable rel. error in the residuals: 0.03
   Max. allowable abs. error in the residuals: 30.0
   Max. allowable rel. error in the r.m.s. of displacement: 0.05
   Max. allowable rel. error in the r.m.s of strain energy: 0.05

2. Perturbation Range:
   +0.1 standard deviation

SOLUTION COMPARISON:

1. Deterministic solution using mean value of radius:

	Stress
Theory	3545.6 psi
NESSUS	3562.2 psi
Difference	0.5%

2. Probabilistic solutions at selected probabilistic levels:

   Theory: 'Exact' CDF based on analytical solution
   NESSUS: Mean-Value-First-Order (MVFO) solution
   Advanced MVFO solution
   (See Figures 2 and 3 for comparison)

REMARKS:

1. For the probabilistic solution of stress (see Figures 2 and 3), a 'calibrated exact' solution was derived by multiplying the theoretica. stresses by a factor of 1.005. This factor is the ratio of the theoretical solution to the NESSUS solution, both computed at the mear values.

2. This validation problem involves only one random variable. In such case, the advanced MVFO solution will yield exact solution. Therefore the difference between the NESSUS solution and the exact solution is c to the finite element solution. However, the error is small (about 1% stress).

3. Figure 3 is the result for the case where the radius has a truncate distribution. This is the reason the resulting probability distributic also is truncated.

Figure V10-2 Case A (Lognormal)

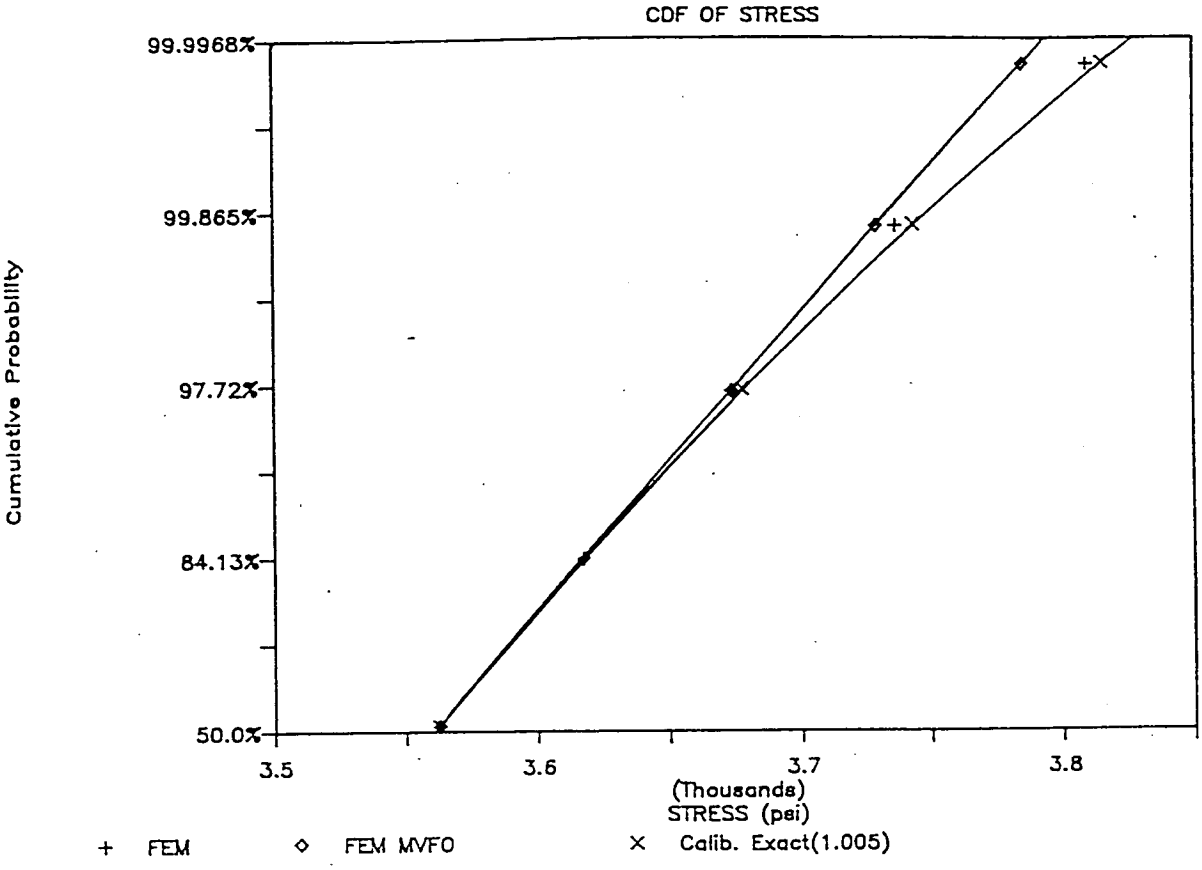
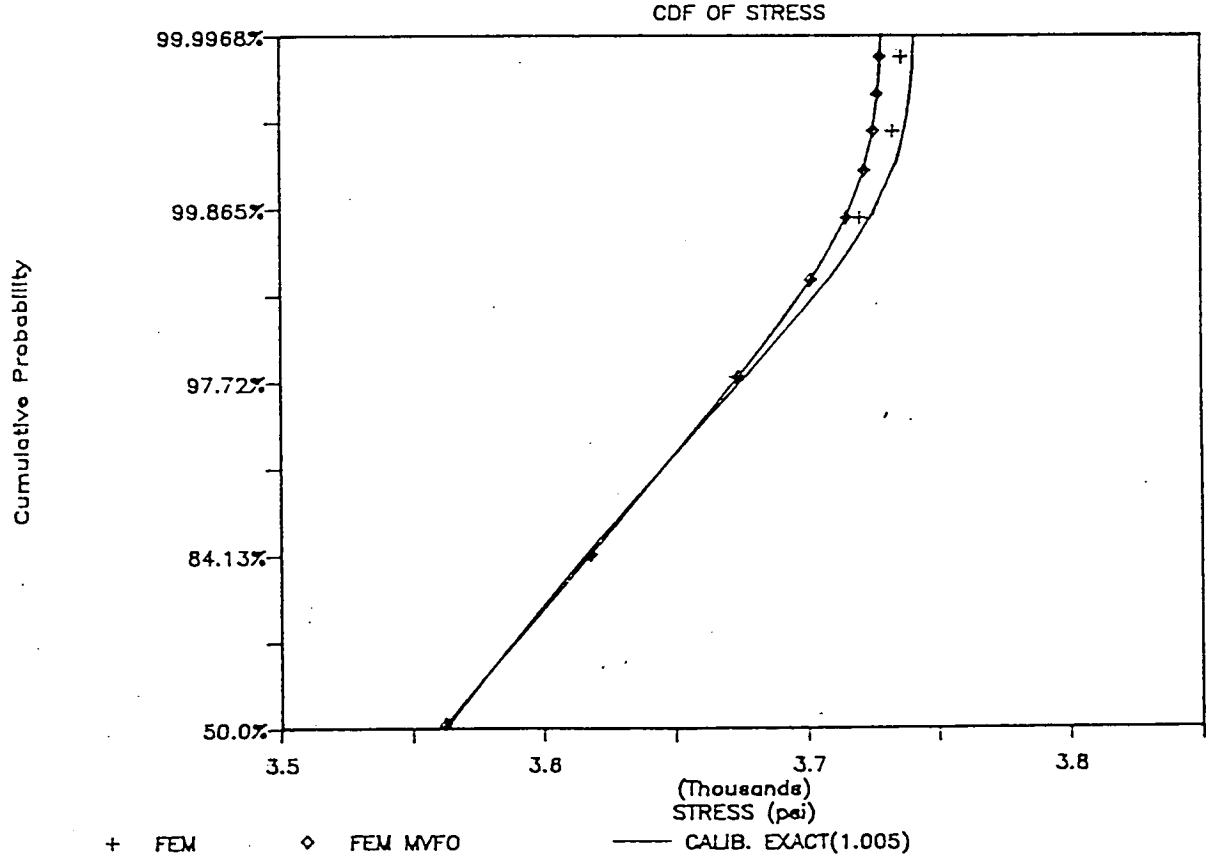


Figure V10-3 Case B (Truncated Normal)



APPENDIX B

A Stress-based Hybrid Finite Element Method for Computational Elasto-plastic Analysis, Using an Endochronic Theory

Prof. Satya Atluri
Mr. Tony Fitzgerald

Georgia Institute of Technology

A STRESS-BASED HYBRID FINITE ELEMENT METHOD FOR COMPUTATIONAL ELASTO-PLASTIC ANALYSIS, USING AN ENDOCHRONIC THEORY

INTRODUCTION

In this section, the hybrid stress finite element will be formulated using the endochronic theory. The iterative scheme for the solution of the nonlinear system of equations that results will be presented with the mid-point radial return algorithm used to improve the accuracy of the integrations.

The motivation for the stress based element is predicated on the observation that the assumed-stress hybrid model has been demonstrated to give more accurate displacements and stress solutions than the conventional displacement model [B.1]. Due to the more accurate stress solution, the use of the hybrid stress model for nonlinear problems, where the nonlinearity arises from the coupling of material behavior to the stress field, should result in a faster convergence.

The use of the classical models of plasticity in a tangent stiffness approach have been reported by Yamada et al [B.2] and Luk [B.3]. Neyssen and Beckers [B.4] reported an increased rate of convergence for a hybrid stress finite element using the classical plasticity theory.

ASSUMPTIONS FOR THE HYBRID FORMULATION

As in the displacement based method, one may assume that the loads and/or displacements are applied incrementally. One must satisfy the following equations within the volume of the element:

Linear Momentum Balance;

$$\Delta \sigma_{ij,j} + \Delta f_i = 0 \qquad \text{in } V_m \qquad (B.1)$$

Angular Momentum Balance;

$$\Delta \sigma_{ij} = \Delta \sigma_{ji} \qquad \text{in } V_m \qquad (B.2)$$

Strain - displacement relation;

$$\Delta \varepsilon_{ij} = \frac{1}{2}(\Delta u_{i,j} + \Delta u_{j,i}) \qquad \text{in } V_m \qquad (B.3)$$

Constitutive relation;

$$\Delta \varepsilon_{ij} = S_{ijkl}\, \Delta \sigma_{kl} \qquad (3.4)$$

Traction Boundary condition;

$$\Delta \sigma_{ij}\, n_j = \Delta \bar{T}_i \qquad \text{at} \quad S_t \qquad (3.5)$$

Displacement Boundary condition;

$$\Delta u_i = \Delta \bar{u}_i \qquad \text{at} \quad S_u \qquad (3.6)$$

Inter-element Boundary conditions;

traction reciprocity

$$\Delta \sigma_{ij}\, n_i^{+} + \Delta \sigma_{ij}\, n_i^{-} = 0 \qquad \text{at} \quad S_o \qquad (3.7)$$

displacement

$$u_i^{+} = u_i^{-} \qquad \text{at} \quad S_o \qquad (3.8)$$

In the above, S_o is defined to be on the boundary of the interface of two elements with the total boundary of an element defined as

$$S = S_p \cup S_\sigma \cup S_u \qquad (3.9)$$

For the derivation of the element stiffness matrix one may assume the following conditions hold a priori. For now, neglect body force, and assume a stress field $\Delta\sigma_{ij}$ which is selected to satisfy the angular momentum balance, $\Delta\sigma_{ij} = \Delta\sigma_{ji}$, and the linear momentum balance $\Delta\sigma_{ij,j} = 0$. Likewise, assume that the change in strain can be related to the change in stress through:

$$\Delta \varepsilon_{ij} = S_{ijkl}\, \Delta \sigma_{kl} \qquad (3.10)$$

Note that in what follows S_{ijkl} was assumed to be composed of an elastic part

and a plastic part with

$$\Delta \varepsilon_{ij} = \Delta \varepsilon^{e}_{ij} + \Delta \varepsilon^{P}_{ij} \qquad (B.11)$$

$$\Delta \varepsilon_{ij} = S^{e}_{ijkl}\, \Delta \sigma_{kl} + S^{P}_{ijkl}\, \Delta \sigma_{kl} \qquad (B.12)$$

where

$$S^{e}_{ijkl} = \frac{\delta_{ik}\,\delta_{jl}}{2\mu} + \frac{\lambda}{(3\lambda+2\mu)2\mu}\,\delta_{ij}\,\delta_{kl} \qquad (B.13)$$

$$S^{P}_{ijkl} = \frac{N_{ij}\,N_{kl}}{(2\mu)(c-1)} \qquad (B.14)$$

WEAK FORM

The following relations must be enforced through the variational statement:

1) compatibility

$$\Delta \varepsilon_{ij} = \frac{1}{2}\left(\Delta u_{i,j} + \Delta u_{j,i}\right) \qquad (B.15)$$

2) traction reciprocity

$$(\Delta \sigma_{ij}\, n_i)^{+} + (\Delta \sigma_{ij}\, n_i)^{-} = 0 \qquad \text{on} \quad S_{\sigma_m} \qquad (B.16)$$

3) traction boundary condition

$$\Delta \sigma_{ij}\, n_i = \Delta \bar{T}_i \qquad \text{on} \quad S_{\sigma_m} \qquad (B.17)$$

4) displacement boundary condition

$$\Delta u_i = \Delta \bar{u}_i \qquad \text{at} \quad S_{u_m}. \qquad (B.18)$$

Defining $\delta \underline{\sigma}$ to be a test function (weighting function) for the compatibility equation (eq. B.15), the weighted form of the compatibility relation becomes:

$$\sum_m \int_{V_m} \left\{ \Delta \varepsilon_{ij} - \frac{1}{2}\left(\Delta u_{i,j} + \Delta u_{j,i}\right) \right\} \delta \sigma_{ij}\, dV = 0 \qquad (B.19)$$

To enforce the traction boundary condition one may use a test function of the same class as the displacemants. Let δu_i be the weighting function for the enforcement of the traction boundary condition. Weighting the traction boundary condition with the test function gives:

$$\sum_m \int_{S_\sigma} (\Delta \sigma_{ij} n_j - \Delta T_i) \delta u_i \, dV = 0 \qquad (B.20)$$

To enforce the traction reciprocity in weak form, one may use the weighting functin δu_i to get

$$\int_{S_\ell} (\Delta \sigma_{ij} n_j^+ + \Delta \sigma_{ij} n_j^-) \delta u_i \, dS = 0 \qquad (B.21)$$

Here $\Delta \sigma_{ij} n_j^+$ represents the traction on one face of an element and $\Delta \sigma_{ij} n_j^-$ represents the traction of the corresponding face of an adjacent element. When summed over all elements, the above reduced to the single term

$$\sum_m \int_{S_\ell} \Delta \sigma_{ij} n_j \delta u_i \, dS = 0 \qquad (B.22)$$

Assuming that the constitutive relations are satisfied a priori, one may write the weak form of the compatibility condition (B.21) as:

$$\sum_m \int_V \{\tfrac{1}{2}(\Delta u_{i,j} + \Delta u_{j,i}) - S_{ijkl} \Delta \sigma_{kl}\} \delta \sigma_{ij} \, dV = 0 \qquad (B.23)$$

Choosing the stress polynomials in such a way that the equilibrium equation was satisfied by the test function allowed one to rewrite the combined weak form as

$$\sum_m \int_{V_m} S_{ijkl} \Delta \sigma_{kl} \delta \sigma_{ij} + \int_{\delta V_m} \delta \sigma_{ij} n_j \Delta u_i + \int_{\delta V_m} \Delta \sigma_{ij} n_j \delta u_i$$

$$- \int_{S_{\sigma m}} \Delta T_i \delta u_i \, dS = 0 \qquad (B.24)$$

DISCRETE WEAK FORM

The stresses within an element were represented as a summation of equilibrate polynomial stress modes, A, with undetermined parameters $\Delta\beta$;

$$\Delta \sigma_{ij} = A_{ijm} \Delta \beta_m \qquad (B.25)$$

Refer to Appendix B for the exact form of the polynomials used to formulate the

hybrid element. To enforce compatibility in a weak form, one may use a test function of the same class as the function for stress. Define $\delta \underline{\sigma}$ as the test function in terms of the same polynomial stress modes, A_{ijm}, with the arbitrary parameters

$$\delta \sigma_{ij} = A_{ijm} \, \delta \beta_m \tag{B.26}$$

The displacements were interpolated from the nodal values, Δq_{fk}, and the standard isoparametric shape functions, ψ_{ik}, as:

$$\Delta u_i = \psi_{ik} \, \Delta q_k \tag{B.27}$$

The trial functions for the displacements and the stresses were expressed in terms of the parameters $\Delta \underline{\beta}$ and $\Delta \underline{q}_f$. Define

$$\delta u_i = \psi_{ik} \, \delta q_k \tag{B.28}$$

as the test function in terms of the shape functions used in the interpolation of the displacements. The parameters, $\delta \underline{q}_f$, will be arbitrary except on the portion of the boundary where the displacements are prescribed, in which case they will be zero.

Substitution of the discrete form for the test and trial functions (Eq. B.25-B.28) expressed the combined weak form in terms of $\Delta \underline{\beta}$ and $\Delta \underline{q}_f$ to give:

$$\sum_m \int_{V_m} \Delta\beta_m A_{ijm} S_{ijkl} A_{kln} \delta\beta_n \, dV + \int_S \Delta q_{fk} \psi_{ik} n_j A_{ijm} \delta\beta_m \, dS +$$

$$\int_S \Delta\beta_m A_{ijm} n_j \psi_{ik} \delta q_{fk} \, dS + \int_{S_{\delta m}} - \Delta T_i \psi_{ik} \delta q_{fk} \, dS = 0 \tag{B.29}$$

Defining the matrix

$$\underline{\underline{H}} = \int_{V_m} A_{ijm} S_{ijkl} A_{kln} \, dV \tag{B.30}$$

and the matrix

$$\underline{\underline{G}} = \int_S \psi_{ik} n_j A_{ijm} \, dS \tag{B.31}$$

One may express the combined weak form in matrix form as

$$\sum_m - \Delta\underline{\beta} \, \underline{\underline{H}} \, \delta\underline{\beta} + \Delta \underline{q}_f \, \underline{\underline{G}} \, \delta\underline{\beta} + \delta \underline{q}_f \, \underline{\underline{G}} \, \Delta\underline{\beta} = \int_S T_i \psi_{ik} \, \delta q_{fk} \tag{B.32}$$

The global stiffness matrix may be assembled with $\Delta\underline{\beta}$ and $\Delta\underline{q}$ retained as unknowns. The number of unknowns at the global level may be reduced by eliminating the stress parameters are assumed only within the domain of the element, with no coupling between elements. For arbitrary $\delta\underline{\beta}$ one must satisfy

$$-\underline{\underline{H}}\,\Delta\underline{\beta} + \Delta\underline{q}\,\underline{\underline{G}} = 0 \qquad (B.33)$$

Thus, $\Delta\underline{\beta}$ may be expressed in terms of the displacement of the element as

$$\Delta\underline{\beta} = \underline{\underline{H}}^{-1}\,\underline{\underline{G}}\,\Delta\underline{q} \qquad (B.34)$$

For arbitrary $\Delta\underline{q}$, then one must have at the global level

$$\sum_m \underline{\underline{G}}\,\Delta\underline{\beta} = \int_{S_\sigma} T_i\,\psi_{ik}\,dS \;\rightarrow\; \sum_m \underline{\underline{G}}\,\underline{\underline{H}}^{-1}\,\underline{\underline{G}}\,\Delta\underline{q} = \int_{S_\sigma} T_i\,\psi_{ik}\,dS \qquad (B.35\ a,b)$$

From the global stiffness matrix that results, $\Delta\underline{q}$ is obtained, with stress parameters computed from equation (B.34).

Relaxing the requirement that the stress field withinthe element satisfy the equilibrium equation a priori allows one to introduce a prescribed body force, f_i. If the linear momentum balance conditions are relaxed and expressed in weak form through the weighting function δu_i the weak form becomes:

$$\sum_m \int_{V_m} (\Delta\sigma_{ij,j} + \Delta f_i)\,\delta u_i\,dV = 0 \qquad (B.36)$$

One has by adding to the combined weak form:

$$\sum_m \int_{V_m} \left\{ (S_{ijkl}\,\Delta\sigma_{kl} - \tfrac{1}{2}(\Delta u_{i,j} + \Delta u_{j,i}) \right\} \delta\sigma_{ij}\,dV \qquad (B.37)$$
$$+ \int_{S_\sigma} (\Delta\sigma_{ij}\,n_j - \Delta T_i)\,\delta u_i\,dS + \int_{S_\sigma} \Delta\sigma_{ij}\,n_i\,\delta u_i\,dS$$
$$+ \int_{V_m} (\Delta\sigma_{ij,j} + \Delta f_i)\,\delta u_i\,dV = 0$$

Which after applying Green's theorem reduces to:

$$\sum_m \int_{V_m} S_{ijkl}\,\Delta\sigma_{kl}\,\delta\sigma_{ij} + \int \Delta u_{i,j}\,\delta\sigma_{ij}\,dV + \int \delta u_{i,j}\,\Delta\sigma_{ij}\,dV$$
$$+ \int_{S_\sigma} \Delta T_i\,\delta u_i\,dS + \int_{V_m} \Delta f_i\,\delta u_i\,dV = 0 \qquad (B.38)$$

After substitution of the discrete form of the trial and test function one may express the combined weak form as:

$$\sum_{m} \Delta\underline{\beta}\, \underline{\underline{H}}\, \delta\underline{\beta} + \Delta\underline{q}_{\gamma}\, \underline{\underline{G}}\, \delta\underline{\beta} + \delta\underline{q}_{\gamma}\, \underline{\underline{G}}\, \Delta\underline{\beta} = \Delta\underline{T}\, \delta\underline{q}_{\gamma} + \Delta\underline{F}\, \delta\underline{q}_{\gamma} \qquad (3.39a)$$

where $$\Delta\underline{T} = \int \Delta T_i\, \psi_{ik}\, dS \; ; \; \Delta\underline{F} = \int \Delta f_i\, \psi_{ik}\, dV \qquad (3.39b)$$

Note that the above formulation, while possessing the same number of unknowns as the weak form where the stress was equilibrated (Eq. 3.39),was substantially less costly to implement numerically. The saving came from the volumetric integration to formulate the matrix $\underline{\underline{G}}$. Performing a volumetric integration allowed for the evaluation of the $\underline{\underline{G}}$ matrix at the same time as the volumetric integration for the matrix $\underline{\underline{H}}$. This means the same Gauss points may be used for the integration of $\underline{\underline{H}}$ and $\underline{\underline{G}}$, as opposed to having to define Gauss points within the volume for $\underline{\underline{H}}$ as well as on the surface for $\underline{\underline{G}}$.

RESIDUAL CALCULATION AND ITERATION SCHEME

The tangent stiffness matrix expressed in equation (B.35) allowed one to compute the change in stress and the change in displacement based on the material parameters at state n being approximately constant over the increment. Due to the linearization of the material behavior, the actual state of stress and the actual displacements at state n deviated from the nonlinear path that material should have followed.

In addition to the errors introduced in the linearization of the material parameters, other errors are generated. For example, if one assumes that the behavior was plastic (perhaps the last load increment caused plastic deformation) but , the next loading increment unloaded the point from the yield surface, then the wrong tangent to the stress-strain relation would have been taken. Likewise, if the material was near the yield point, and the next loading increment caused it to go from elastic material behavior to plastic behavior, then the assumption of elastic material behavior for the increment would not be valid over the entire increment.

For the case where a plastic stress/strain increment was assumed, but the resulting load increment moved the stress point to inside the yield surface, one was left with no alternative but to reformulate the stiffness matrix to reflect the correct tangent to the material properties. One must then recalculate the change in stress based on the correct stress-strain path. For the case where the path was part elastic and part plastic one must divide the

stress into two parts, applying the first part as an elastic process, then all the second part to be applied in the residual calculation.

To illustrate the part elastic-part plastic case, assume that at sate n, no plastic strain has occured. Let the next load increment be such that only a part of the stress increment may be applied elastically. For a given change in body forces and change in surface tractions assume that the corresponding change in stress predicted by the elastic stiffness formulation is such that

$$F^{*} = (\underset{\sim}{\sigma}'_A + \Delta\underset{\sim}{\sigma}' - \underset{\sim}{\alpha}^{n}) : (\underset{\sim}{\sigma}'_A + \Delta\underset{\sim}{\sigma}' - \underset{\sim}{\alpha}^{n}) - R^2 > 0 \qquad (B.40)$$

i.e. the stress point, if elastic material behavior is assumed, would fall to the outside of the yield surface. At a point, assume that the stress $\underset{\sim}{\sigma}_A$ lies o the inside of the yield surface. Let $\underset{\sim}{\sigma}_B$ be the point on the yield surface wher the trajectory of $\underset{\sim}{\sigma}_A + \Delta\underset{\sim}{\sigma}$ intersects the yield surface. The point of intersection may be computed as

$$\underset{\sim}{\sigma}_B = \underset{\sim}{\sigma}_A + r\,\Delta\underset{\sim}{\sigma} \qquad (B.41)$$

where

$$r = \frac{\bar{\sigma}_A - \bar{\sigma}_Y}{\bar{\sigma}_A - (\sigma_A + \Delta\sigma)} \qquad (B.42)$$

Only the portion of $\Delta\underset{\sim}{\sigma}$ that is required to move the stress point to the yiel surface is applied, with the rest of the stress that would occur during the plastic material behavior neglected for now. The resulting stress field will not be in equilibrium; however, one may compute the out-of-balance loads neede to produce an equilibrated stress field at state n+1. For equilibrium at n+1 one should have:

$$\sigma^{n+1}_{ij,j} + f^{n+1}_{i} = 0 \qquad (B.43\,a)$$

$$\sigma^{n+1}_{ij}\, n_j = T^{n+1}_{i} \qquad (B.43\,b)$$

Weighting the above with the test function δu_i will give after application of Green's theorem:

$$R_{\sigma_k}\,\delta q_k = \int_{V_m} \sigma^{n+1}_{ij}\,\psi_{ik,j}\,\delta q_k\, dV \qquad (B.44)$$

The out-of-balance loads will be:

$$Q^{i}_{k} = \int T^{n+1}_{i}\,\psi_{ik}\, dS + \int f^{n+1}_{j}\,\psi_{ik}\, dV - R^{i}_{\sigma_k} \qquad (B.45)$$

For the points where the elastically applied stress exceeded the yield stress, the process should be plastic. The stiffness matrix may be updated to reflect the plastic process and allow the out-of-balance loads to follow the plastic stress-strain path. This will give a correction to the displacements as

$$\Delta \underline{q}^{i} = \underline{\underline{G}}\,\underline{\underline{H}}^{-1}\,\underline{\underline{G}}\,\underline{Q}^{i} \qquad (B.46)$$

and a correction to the stress of

$$\Delta \sigma_{ij}^{i} = A_{ijm}\,\Delta \beta_{m}^{i} \quad ; \quad \Delta \beta_{m}^{i} = \underline{\underline{G}}\,\underline{\underline{H}}^{-1}\,\Delta \underline{q} \qquad (B.47\ a, b)$$

with the stress at n+1 being given by:

$$\underset{\sim}{\sigma}^{n+1} = \underset{\sim}{\sigma}^{n} + \Delta \underset{\sim}{\sigma}_{AB} + \sum_{i} \Delta \underset{\sim}{\sigma}^{i} \qquad (B.48)$$

One may compute the strain at n+1 from the stress using the constitutive relation as:

$$\underset{\sim}{\varepsilon}^{n+1} = \underset{\sim}{\varepsilon}^{n} + \underline{\underline{S}}^{e}\left(\Delta \underset{\sim}{\sigma}_{AB} + \Delta \underset{\sim}{\sigma}_{BC}\right) + \underline{\underline{S}}^{p}\,\Delta \underset{\sim}{\sigma}_{BC} \qquad (B.49)$$

The displacements at n+1 will be given as

$$\underline{q}^{n+1} = \underline{q}^{n} + \Delta \underline{q} + \sum_{i} \Delta \underline{q}^{i} \qquad (B.50)$$

Due to the above approach of splitting the stress into two parts, there will be errors in compatibility. At state n+1 one should have

$$\varepsilon_{ij}^{n+1} - \frac{1}{2}\left(u_{i,j} + u_{j,i}\right)^{n+1} = 0 \qquad (B.51)$$

where ε_{ij} is computed from the stress through the constitutive relation.

To enforce this condition, a weighting function of the same class as the function for the stress field may be used. The following load due to the error in compatibility is obtained:

$$\underline{R}_{\varepsilon}\,\delta \underline{\beta} = \int_{V_m} \left\{ \varepsilon_{ij}^{n+1} - \frac{1}{2}\left(u_{i,j} + u_{j,i}\right)^{n+1} \right\} \delta \sigma_{ij}\, dV \qquad (B.52)$$

Application of the above residual to the system will give displacements that are compatible with the strains obtained from the stress field. There will be some redistribution of the stress when the strain residual is applied, but for

the most part, the displacements will change more during each iteration than the stress. One may apply both $\underline{R}_\sigma$ and $\underline{R}_\varepsilon$ at the same time, and continue the iteration process until the norm of the displacement does not change significantly.

CONSISTENCY CONDITION

With the above hybrid method, unless the stress/strain increments are ver small, there will be errors in the consistency condition.

$$F^{n+1} \equiv (\underset{\sim}{\sigma}^{n+1} - \underset{\sim}{\alpha}^{n+1}) : (\underset{\sim}{\sigma}^{n+1} - \underset{\sim}{\alpha}^{n+1}) - R^2 \neq 0 \qquad (B.53)$$

The errors may be reduced by using a mid-point rule for the integration of the strains. the plastic strain may be computed from the stress as:

$$\Delta \underset{\sim}{\varepsilon}^p = \int_{\sigma^n}^{\sigma^{n+1}} \underset{\sim}{N} \frac{\underset{\sim}{N} : d\underset{\sim}{\sigma}}{2\mu(C-1)} \qquad (B.54)$$

For the finite change $\Delta\underset{\sim}{\sigma}$ one may use the approximation

$$\Delta \underset{\sim}{\varepsilon}^p = \underset{\sim}{N}^* \frac{\underset{\sim}{N}^* : \Delta\underset{\sim}{\sigma}}{2\mu(C-1)} \qquad (B.55)$$

where

$$\underset{\sim}{N}^* = \frac{((\underset{\sim}{\sigma}^n + \beta\Delta\underset{\sim}{\sigma}) - (\underset{\sim}{\alpha}^n + \beta\Delta\underset{\sim}{\alpha}))}{\|(\underset{\sim}{\sigma}^n + \beta\Delta\underset{\sim}{\sigma}) - (\underset{\sim}{\alpha}^n + \beta\Delta\underset{\sim}{\alpha})\|} \qquad (B.56)$$

Likewise, the change in strain for a plastic process may be approximated as:

$$\Delta\underset{\sim}{\varepsilon} = \frac{\Delta\underset{\sim}{\sigma}}{2\mu} + \frac{\lambda}{(3\lambda+2\mu)2\mu}(\Delta\underset{\sim}{\sigma} : \underset{\sim}{I})\underset{\sim}{I} + \frac{\underset{\sim}{N}^*}{2\mu}\frac{\underset{\sim}{N}^* : \Delta\underset{\sim}{\sigma}}{C-1} \qquad (B.57)$$

Using the mid-point rule will lead to compatibility errors. However, the application of the residual $\underline{R}_\varepsilon$ will correct the errors that accumulate due to compatibility.

The final system of equations that result when both $\underline{R}_\varepsilon$ and $\underline{R}_\sigma$ are applie during the iteration process will have the form

$$\begin{bmatrix} -\underset{=}{H} & \underset{=}{G} \\ \underset{=}{G} & \underset{=}{0} \end{bmatrix} \times \begin{Bmatrix} \Delta\underline{\beta} \\ \Delta\underline{q} \end{Bmatrix} = \begin{Bmatrix} \underline{R}_\varepsilon \\ \underline{R}_\sigma + \Delta\underline{T} + \Delta\underline{F} \end{Bmatrix} \qquad (B.58)$$

Here, the matrix $\underline{\underline{G}}$ is constant and only need be evaluated once. The matrix $\underline{\underline{H}}$ depends on the material behavior, and must be evaluated for each iteration.

As each iteration, i, is carried out, the stress and displacement is updated as:

$$\underset{\sim}{\sigma}^{n+1} = \underset{\sim}{\sigma}^{n} + \Delta \underset{\sim}{\sigma}_{AB} + \sum_{i} \Delta \underset{\sim}{\sigma}^{i} \qquad (B.59)$$

$$\underline{q}^{n+1} = \underline{q}^{n} + \Delta \underline{q} + \sum_{i} \Delta \underline{q}^{i} \qquad (B.60)$$

The strain must be computed in two parts with the elastic part given by

$$\Delta \underset{\sim}{\varepsilon}_{AB} = \underset{\approx}{S}^{e} \Delta \underset{\sim}{\sigma}_{AB} \qquad (B.61)$$

and the plastic part by

$$\Delta \underset{\sim}{\varepsilon}_{BC} = \underset{\approx}{S}^{e} \Delta \underset{\sim}{\sigma}^{i} + \underset{\approx}{S}^{p} \sum_{i} \Delta \underset{\sim}{\sigma}^{i} \qquad (B.62)$$

CONSTITUTIVE EQUATIONS

The equations used to characterize the behavior of a material and its reaction to applied loads are called constitutive equations, since they describe the macroscopic behavior resulting from the internal constitution of the material. The objective of a constitutive relation is to provide a good description of the relationship between stress and strain for a given material. The problem is complicated by the fact that different classes of materials exhibit different characteristics. The goal of well-formulated constitutive theory is to allow for all of the different observed phenomena to be described by the same mathematical formulation.

The mathematical model governing the elastic-plastic behavior of solids, in particular, should have the following key ingredients: i) a relationship between stress and strain to describe the behavior under elastic conditions; ii) a criterion which will indicate the level of stress at which plastic strains will occur; iii) laws governing the growth of plastic strains as the material is stressed/strained beyond the elastic range; iv) laws governing the change in elastic limit as plasticity develops (strain harding, Baushinger effect, strain softening).

The general theory of internal variables has played a key role in the development of more and more realistic constitutive models to characterize inelastic material behavior. Typical internal variables that are widely employed include i) the so-called 'back-stress' (the tensor locating the center of the yield surface in stress-space), ii) the parameters that

characterize the expansion of the yield surface, iii) the parameters that characterize the 'bounding-surface' in multi-yield-surface theories of plasticity [B.5-B.9], iv) the 'drag-stress' used to characterize the creep surface.

Of the constitutive relations proposed for inelasticity, the 'internal-time' (endochronic) theory of Valanis[B.10], Watanabe and Atluri [B.11], The Multi-yield-surface theories of Morz [B.5, B.6], Dafalias & Popov [B.7, B.8], Kreig [B.9], and the internal variable theories of Onat [B.12, B.13, B.14], Fardshishen & Onat [B.15], Onat & Fardshisheh [B.16], Chaboche [B.17], Chaboche & Rousselier [B.18] all appear on the surface to be unrelated to each other and to be based on totally diverse concepts. The work of Watanabe & Atluri [B.19] places all the relations in perspective by showing that the 'internal-time' theory [B.10, B.11] is general enough to encompass all other relations reported in the literature as special cases. Likewise, [B.19] shows that their internal time theory as expressed in differential form is no more difficult to implement numerical than the classical Prager-Ziegler kinematic harding theory.

The 'Endochronic Theory' was presented by Valanis [B.20, B.21] in 1971 and held out the prospect of explaining the experimentally observed phenomena of cross-hardening, cyclic hardening, and initial strain problems. While the initial version of the theory was subject to much criticism [B.22], certain features of the theory allow for constitutive laws that are better in modeling observed phenomena in cyclic plasticity of metals than the classical elasto-plastic constitutive relations.

The new intrinsic time model presented by Valanis [B.23] in 1980 rectifie some of the shortcomings of the earlier theory. The work of Valanis and Fan [B.24] presented an incremental or differential form of the integral relation of stress and strain for plasticity [B.23]. The computational implementation of the differential relation in [B.24] is not in a standard 'tangent stiffness format, thus, a finite element formulation in thetraditional sense is not possible. Watanabe & Atluri [B.11]present an alternative derivation of the differential stress-strain relation using the concept of intrinsic time dependent on plastic strain [B.23]. This alternative derivation presents the endochronic theory in a structure that is similar to that of classical plasticity, thus, leading to a stiffness type finite element formulation.

While the endochronic relation as developed by Watanabe &Atluri [B.11] is similar in its structure to that of classical plasticity, there are several novel advantages present in the endochronic theory not present in the classical plasticity theory. The ability to model test data for both monotonic or cyclic plasticity as accurately as possible, with a minimal number of material parameters makes the endochronic theory a simple theory to implement in a finite element code.

Summary of the Endochronic theory

The deviatoric stress, $\underline{\sigma}'$, is related to the mean stress, σ_m, by

$$\sigma'_{ij} = \sigma_{ij} - \sigma_m \delta_{ij} \equiv S_{ij} \qquad (B.63)$$

with

$$\sigma_m = \delta_{ij}\,\sigma_{ij}/3 \tag{B.64}$$

The back-stress, $\underset{\sim}{\alpha}$, is defined as the center of the yield surfsce in deviatoric stress space. One may, for the infinitesimal strain problem, let the differential strain tensor, $d\underset{\sim}{\varepsilon}$, be composed of elastic and plastic components,

$$d\underset{\sim}{\varepsilon} = d\underset{\sim}{\varepsilon}^{e} + d\underset{\sim}{\varepsilon}^{p} \tag{B.65}$$

For metals, the assumption of plastic strains being only deviatoric in nature allows one to write the differential of strain as:

$$d\varepsilon_{ij} = de^{e}_{ij} + de^{p}_{ij} + \delta_{ij}\frac{d\varepsilon^{e}_{kk}}{3} \tag{B.66}$$

where $d\underset{\sim}{e}'$ is the deviatoric component of strain.

Following Watanabe and Atluri [B.11], one may define at a material point, an intrinsic time measure, ζ, related to the magnitude of plastic strain that has accumulated at that point as:

$$d\zeta = \left(d\underset{\sim}{e}^{p} : d\underset{\sim}{e}^{p}\right)^{1/2} \tag{B.67}$$

As in the classical theories of plasticity, the isotropic expansion of the yield surface is asumed to be a function of the magnitude of plastic strain. The isotropic expansion is introduced though the non-negative function

$$f(\zeta) \quad \text{with} \quad f(0) = 1. \tag{B.68}$$

A differential intrinsic time,

$$dz = \frac{d\zeta}{f(\zeta)} \tag{B.69}$$

is defined from the magnitude of plastic strain and the function describing the growth of the yield surface. From Valanis [B.10], and modified by Watanabe and Atluri [B.11], let the deviatoric stress be related to the plastic strain through

$$\underset{\sim}{\sigma}' = \underset{\sim}{S} = 2\mu \int_0^z \rho(z-z')\frac{\partial \underset{\sim}{e}^{p}}{\partial z'}\,dz' \tag{B.70}$$

In order to recover a yield surface, allow the kernal, $\rho(z)$, to be of the singular form:

$$\rho(z) = \rho_0(z) + \rho_1(z) \tag{B.71}$$

where, $\delta(z)$ is a Dirac delta function and $\rho_1(z)$ is a non-singular function, Substitution of (B.71) into (B.70) results in the deviatoric stress being related through the equation as

$$\underset{\sim}{S} = 2\mu\,\rho_0 \frac{d\underset{\sim}{e}^p}{dz} + \underset{\sim}{\alpha}(z) \tag{B.72}$$

with

$$\alpha(z) = 2\mu \int_0^z \rho_1(z-z') \frac{\partial \underset{\sim}{e}^p}{\partial z'}\, dz' \tag{B.73}$$

to give

$$\frac{d\underset{\sim}{\xi}^p}{(d\underset{\sim}{\xi}^p : d\underset{\sim}{\xi}^p)} = \frac{\underset{\sim}{S} - \underset{\sim}{\alpha}}{2\mu\,\rho_0 f(\zeta)} \tag{B.74}$$

Let $S_Y^0 = 2\mu\rho_0$ be the initial radius of a yield surface, and let be the radius of the yield surface as plasticity develops.

In order to distinguish an elastic process from a plastic process, one may look at the conditions required for $d\zeta$ to be non-zero. From the definition of the differential intrinsic time measure the magnitude of plastic strain is expressed as:

$$d\zeta = \left[d\underset{\sim}{\xi}^p : d\underset{\sim}{\xi}^p \right]^{1/2} \tag{B.75}$$

During plastic flow, from the definition of the direction of plastic strain, one has, by definition

$$\frac{d\underset{\sim}{\xi}^p}{d\zeta} : \frac{d\underset{\sim}{\xi}^p}{d\zeta} = 1 \tag{B.76}$$

or

$$\frac{S-\alpha}{f(\zeta) S_Y^0} : \frac{S-\alpha}{f(\zeta) S_Y^0} = 1 \tag{B.77}$$

or

$$[\underset{\sim}{S} - \underset{\sim}{\alpha}] : [\underset{\sim}{S} - \underset{\sim}{\alpha}] = R(\zeta)^2 \qquad (B.78)$$

Equation (B.78) may be viewed as the equivalent of the classical Von Mises yield criterion. Thus, if plastic flow occurred during an increment of stress/strain, the above equations (B.76-B.78) should be satisfied throughout, and at the end of the increment.

The direction of plastic strain is given by

$$\frac{d\underset{\sim}{\varepsilon}^p}{d\zeta} = \underset{\sim}{N} = \frac{\underset{\sim}{S} - \underset{\sim}{\alpha}}{S_y^o \, f(\zeta)} \qquad (B.79)$$

where

$$d\underset{\sim}{\varepsilon}^p = d\zeta \, \frac{[\underset{\sim}{S} - \underset{\sim}{\alpha}]}{S_y^o \, f(\zeta)} \qquad (B.80)$$

which may be expressed as

$$d\underset{\sim}{\varepsilon}^p = d\zeta \, \underset{\sim}{N} \qquad (B.81)$$

In equation (B.81), the tensor $\underset{\sim}{N}$ is analogous to the normal to the classical yield surface. From the definition of $d\zeta$, for admissible plastic flow, $d\zeta$ must be non-negative.

$$\begin{aligned} d\zeta > 0 \quad & \text{plastic flow admissible} \\ d\zeta < 0 \quad & \text{plastic flow not admissible} \end{aligned} \qquad (B.82)$$

Taking the trace of both sides of Eq. (B.81) with the differential of plastic strain, $d\underset{\sim}{\varepsilon}^p$, gives the requirement of admissible plastic flow in terms of the normal to the yield surface and the plastic strain.

$$d\underset{\sim}{\varepsilon}^p : \underset{\sim}{N} = d\zeta > 0 \qquad (B.83)$$

Equation (B.83) is not a convenient condition to apply within a finite element codes, since the finite element code will return directly $d\underset{\sim}{\varepsilon}$ not $d\underset{\sim}{\varepsilon}^p$. Therefore, the admissible flow condition is best expressed in terms of the differential of total strain, $d\underset{\sim}{\varepsilon}$.

To express d in terms of $d\underset{\sim}{\varepsilon}$ directly requires differentiation of equations (B.80) with respect to $d\zeta$.
This gives:

$$\frac{d^2\underset{\sim}{\xi}^p}{d\zeta^2} f(\zeta) + \frac{d\underset{\sim}{\xi}^p}{d\zeta} f'(\zeta) = \frac{1}{S_y^*}\left(\frac{d\underset{\sim}{S}}{d\zeta} - \frac{d\underset{\sim}{\alpha}}{d\zeta}\right) \tag{B.84}$$

The rate of deviatoric stress with respect to the magnitude of plastic strain, $d\underset{\sim}{S}/d\zeta$, may be obtained by noting that the plastic strain is expressed as

$$d\underset{\sim}{\xi}^p = d\underset{\sim}{\xi}' - \frac{d\underset{\sim}{S}}{2\mu} \tag{B.85}$$

which may be expressed as

$$\frac{d\underset{\sim}{\xi}^p}{d\zeta} = \frac{d\underset{\sim}{\xi}'}{d\zeta} - \frac{d\underset{\sim}{S}}{2\mu\, d\zeta} \tag{B.86}$$

or

$$\frac{d\underset{\sim}{S}}{d\zeta} = 2\mu\left(\frac{d\underset{\sim}{\xi}'}{d\zeta} - \frac{d\underset{\sim}{\xi}^p}{d\zeta}\right) \tag{B.87}$$

The tensor, $\underset{\sim}{\alpha}$, which is analogous to the center of the yield surface in stress space is expressed by the integral

$$\underset{\sim}{\alpha} = 2\mu \int_0^z \rho_1(z - z') \frac{d\underset{\sim}{\xi}^p}{dz'}\, dz' \tag{B.88}$$

Recall that Leibnitz's rule allows one to differentiate under an integral as

$$\frac{d}{dx}\int_A^B f(x,t)\, dt = \int_A^B \frac{\partial f(x,t)}{\partial x}\, dt + f(x,B)\frac{\partial B}{\partial x} - f(x,A)\frac{\partial A}{\partial x} \tag{B.89}$$

One may express the rate of change of the center of the yield surface with respect to the magnitude of plastic strain as

$$\frac{d\underset{\sim}{\alpha}}{d\zeta} = 2\mu\left(\frac{\underset{\sim}{h}^*}{f} + \rho_1(0)\frac{d\underset{\sim}{\xi}^p}{d\zeta}\right) \tag{B.90}$$

where

$$\underset{\sim}{h}^* = \int_0^z \frac{\partial \rho_1(z - z')}{\partial z} \frac{d\underset{\sim}{\xi}^p}{dz'}\, dz' \tag{B.91}$$

Substitution of (B.87) and (B.90) into (B.84) gives after rearrangement:

$$d\underset{\sim}{\xi}' = d\underset{\sim}{\xi}^p\left[1+\rho_1(0) + \frac{S_y^o f'(\zeta)}{2\mu}\right] + \frac{\underset{\sim}{h}^*}{f}\, d\zeta + S_y^o \frac{d^2\underset{\sim}{\xi}^p}{d\zeta^2} f(\zeta)\, d\zeta \tag{B.92}$$

Taking trace of both sides with normal, $\underset{\sim}{N}$, gives

$$d\underset{\sim}{\xi}' : \underset{\sim}{N} = d\zeta\left[1+\rho_1(0) + \frac{S_y^o f'(\zeta)}{2\mu} + \frac{\underset{\sim}{h}^* : \underset{\sim}{N}}{f} + S_y^o \frac{d^2\underset{\sim}{\xi}^p}{d\zeta^2} : \underset{\sim}{N} f\right] \tag{B.93}$$

Note that since

$$\frac{d\underset{\sim}{\xi}^p}{d\zeta} : \frac{d\underset{\sim}{\xi}^p}{d\zeta} = 1 \tag{B.94}$$

it follows that

$$\frac{d^2\underset{\sim}{\xi}^p}{d\zeta^2} : \frac{d\underset{\sim}{\xi}^p}{d\zeta} = \frac{d^2\underset{\sim}{\xi}^p}{d\zeta^2} : N = 0 \tag{B.95}$$

The magnituge of plastic strain expressed in terms of the total strain and the normal to the yield surface is given by:

$$\frac{d\underset{\sim}{\xi}' : \underset{\sim}{N}}{C} = d\zeta \tag{B.96}$$

where

$$C = \left[1+\rho_1(0) + \frac{S_y^o f'(\zeta)}{2\mu} + \frac{\underset{\sim}{h}^* : \underset{\sim}{N}}{f}\right] \tag{B.97}$$

For an increment in total strain $d\underset{\sim}{\xi}$, the criterion for establishing whether or not a process leads to an admissible plastic strain is expressed now in terms of $d\underset{\sim}{\xi}$ instead of $d\underset{\sim}{\xi}p$.

To summarize, for plastic strain to be admissible, the following must hold:

$$(\underset{\sim}{S} - \underset{\sim}{\alpha}) : (\underset{\sim}{S} - \underset{\sim}{\alpha}) = R^2 \tag{B.98}$$

and

$$d\underset{\sim}{\varepsilon}' : \underset{\sim}{N} > 0 \tag{B.99}$$

Otherwise, an elastic process will occur if

$$(\underset{\sim}{S} - \underset{\sim}{\alpha}) : (\underset{\sim}{S} - \underset{\sim}{\alpha}) < R^2 \tag{B.100}$$

or

$$(\underset{\sim}{S} - \underset{\sim}{\alpha}) : (\underset{\sim}{S} - \underset{\sim}{\alpha}) = R^2 \tag{B.101}$$

and

$$d\underset{\sim}{\varepsilon}' : \underset{\sim}{N} \leq 0$$

Here $\underset{\sim}{\alpha}$ is viewed as the back stress, or in the geometric description of the yield surface, as the center of the yield surface in deviatoric stress-space. R is viewed as the radius of the yield surface. Note that when a monotonically increasing load is applied, the stress and the back stress are co-axial, and the simple picture of the yield cylinder moving in stress space is possible. However, in the general case of non-proportional loading, the back stress and the yielding stress are not co-axial and the geometric picture cannot be drawn. In the case where $\underset{\sim}{\sigma}$ and $\underset{\sim}{\alpha}$ are not coaxial, the back stress does not reduce to three principal directions in stress space. Instead, it is composed of six components. One may still get an idea of how the yield surface is translating, if one plots a projection of the yield surface.

An incremental (or rate or differential) form of the stress-strain relation in the presence of plastic deformation is required for formulation of the computational scheme in a variational sense. Recall that the total differential strain is assumed to be made up of an elastic part plus an inelastic part. From Eq. 1.23 the plastic strain is expressed in terms of the total strain and the deviatoric stress as:

$$d\underset{\sim}{\varepsilon}^p = d\varepsilon' - \frac{d\underset{\sim}{S}}{2\mu} \tag{B.102}$$

The plastic strain may be expressed in terms of the magnitude of plastic strain, $d\zeta$, and the normal to the yield surface, $\underset{\sim}{N}$. Using Equation (B.80), (B.96) in (B.102), the deviatoric stress is expressed in terms of the deviatoric strain:

$$\begin{aligned} d\underset{\sim}{S} &= 2\mu\,(d\underset{\sim}{\varepsilon}' - d\zeta\,\underset{\sim}{N}) \\ &= 2\mu\, d\underset{\sim}{\varepsilon}' - 2\mu\, d\underset{\sim}{\varepsilon}^p \frac{\underset{\sim}{N} : d\underset{\sim}{\varepsilon}'}{c}\,\underset{\sim}{N} \end{aligned} \tag{B.103}$$

or

$$d\underset{\sim}{S} = 2\mu\, d\underset{\sim}{\varepsilon}' - \frac{2\mu(\underset{\sim}{S}-\underset{\sim}{\alpha})(\underset{\sim}{S}-\underset{\sim}{\alpha}):d\underset{\sim}{\varepsilon}'}{c\, S_y^{o\,2}\, f(\zeta)^2} \tag{B.104}$$

The total differential of stress becomes:

$$d\underset{\sim}{\sigma} = 2\mu\, d\underset{\sim}{\varepsilon} + \lambda(\underset{\sim}{I}:d\underset{\sim}{\varepsilon})\,\underset{\sim}{I} - \frac{2\mu(\underset{\sim}{S}-\underset{\sim}{\alpha})(\underset{\sim}{S}-\underset{\sim}{\alpha}):d\underset{\sim}{\varepsilon}}{c\, S_y^{o\,2}\, f(\zeta)^2} \tag{B.105}$$

where

$$c = 1 + \rho_1(o) + \frac{(\underset{\sim}{S}-\underset{\sim}{\alpha}):\underset{\sim}{h}^*}{S_y^o\, f(\zeta)} + \frac{S_y^o\, f'(\zeta)}{2\mu}$$

$$\underset{\sim}{h}^* = \int_o^z \frac{\partial \rho_1(z-z')}{d\zeta}\,\frac{d\underset{\sim}{\varepsilon}^p}{dz'}\,dz' \tag{B.106}$$

with the rate form for back stress expressed as:

$$d\underset{\sim}{\alpha} = 2\mu\,\rho_1(o)\,d\underset{\sim}{\varepsilon}^p + 2\mu\,\frac{\underset{\sim}{h}^*}{f(\zeta)}\left(d\underset{\sim}{\varepsilon}^p : d\underset{\sim}{\varepsilon}^p\right)^{1/2} \tag{B.107}$$

Defining the correct form for $\rho_1(z)$ and $f(\zeta)$, allows the yield surface expansion and translation of the yield surface to be prescribed in any manner that one wishes.

The most convenient form for $\rho_1(z)$ is expressed as a sum of exponential terms, such as

$$\rho_1(z) = \sum_i \rho_{1i}\, e^{-a_i z} \tag{B.108}$$

By proper choice of the constants ρ_{1i} and a_i, the rate of kinematic hardening may be controlled by the form of $f(\zeta)$. For linear isotropic hardening, one may use:

$$f(\zeta) = 1 + \beta\,\zeta \tag{B.109}$$

where β is the rate of isotropic expansion of the yield surface. For non-linear isotropic hardening, Watanabe and Atluri have suggested the form

$$f(\zeta) = a + (1-a)\, e^{-\gamma\zeta} \tag{B.110}$$

where γ and a are chosen to fit a given material.

For the exact procedure used to select the constants ρ_{1i}, a_i, and β (or γ, and a) refer to Appendix A where the incremental form for the case of uniaxial tension is expressed in terms of σ_{11} and ε_{11}. While all that is required for determining the constants is the uniaxial tension test, the test must be perfqrmed over several cycles of load so that the hysteresis loops of stress/strain are available. This cycling is needed to separate the Baushinger effect from the isotropic expansion.

With $\rho_1(z)$ expressed as an exponential form as in (B.108), the rate form of the endochronic theory so described reduces to

$$d\sigma_{ij} = E_{ijkl}\; d\varepsilon_{kl} \tag{B.111}$$

with

$$E_{ijkl} = 2\mu\, \delta_{ik}\, \delta_{jl} + \lambda\, \delta_{ij}\, \delta_{kl} - \frac{2\mu}{c}\, N_{ij}\, N_{kl}\, \Gamma \tag{B.112}$$

and

$$N_{ij} = \frac{S_{ij} - \alpha_{ij}}{R(\zeta)} \tag{B.113}$$

with $\Gamma=1$ if the increment in strain results in a plastic process or $\Gamma=0$ if the increment is an elastic process. Here $\underset{\sim}{N}$ may be viewed as the normal to a yield surface in stress space as in classical plasticity. The rate of growth of the back stress is given by

$$d\underset{\sim}{\alpha} = \sum_i 2\mu \int_0^z -a_i\, \rho_{1i}\, e^{-a_i(z-z')}\, \frac{d\underset{\sim}{\varepsilon}^p}{dz'}\, dz' \tag{B.114}$$

or

$$d\underset{\sim}{\alpha} = \sum_i d\underset{\sim}{\alpha}^i \;; \qquad d\underset{\sim}{\alpha}^i = 2\mu\, \rho_{1i}\, d\underset{\sim}{\varepsilon}^p - \underset{\sim}{\alpha}^i\, a_i\, \frac{d\zeta}{f(\zeta)} \tag{B.115}$$

with the plastic strain given by

$$d\underset{\sim}{\varepsilon}^p = \frac{1}{c}\, \underset{\sim}{N}\, (\underset{\sim}{N} : d\underset{\sim}{\varepsilon}) \tag{B.116}$$

Note that the endochronic theory departs from the classical plasticity theory in the sense that the back stress is an assumed quantity in the

classical development. Whereas in the endochronic theory, the evolution of the back stress arises from the assumption of the stress being related to an intrinsic time measure. The rate form of the endochronic theory is summarized in the table below:

Table 1 : Summary of the Internal-Time Theory of Plasticity

Endochronic Theory :

$$d\sigma_{kk} = (2\mu + 3\lambda)\, d\varepsilon_{kk}\ (\underset{\sim}{\varepsilon})$$

where μ , λ are lame constants

$f(\zeta) = 1 + \beta\zeta$ (linear); or

$f(\zeta) = a + (1-a)\, e^{-r\zeta}$ (exponential)

$$C = 1 + \rho_1(0) + \frac{(\underset{\sim}{s} - \underset{\sim}{\alpha}) : \underset{\sim}{h}^*}{s_y^\circ\, f^2(\zeta)} + \frac{s_y^\circ \left(df/d\zeta\right)}{2\mu}$$

$$s_y^\circ = 2\mu\, \rho_0$$

$$\rho(z) \equiv \rho_0\, \delta(z) + \rho_1(z) + \sum_i \rho_{1i}\, e^{-a_i z}$$

$$\underset{\sim}{\alpha} = \sum_i \underset{\sim}{\alpha}_i^{(i)} \quad ; \quad \underset{\sim}{h}^* = \sum_i \underset{\sim}{h}^{*(i)} = \sum_i -\frac{a_i}{2\mu}\, \underset{\sim}{\alpha}^{(i)}$$

Rate of Kinematic Harding:

$$d\underset{\sim}{\alpha}^{(i)} = 2\mu\, \rho_{1i}\, d\underset{\sim}{\varepsilon}^p - \frac{a_i\, \underset{\sim}{\alpha}^{(i)}}{f} \left(d\underset{\sim}{e}^p : d\underset{\sim}{e}^p\right)^{1/2}$$

(no sum on i) for $i = 1, 2 \ldots$

$$d\underset{\sim}{\alpha} = \sum_i d\underset{\sim}{\alpha}^{(i)} = 2\mu \sum_i \rho_{1i}\, d\underset{\sim}{e}^p - \sum_i \frac{a_i\, \underset{\sim}{\alpha}^{(i)}}{f} \left(d\underset{\sim}{\varepsilon}^p : d\underset{\sim}{\varepsilon}^p\right)^{1/2}$$

Rate of Isotropic Hardening:

(linear f)

$$ds = s_y^\circ\, \beta \left[d\underset{\sim}{e}^p : d\underset{\sim}{e}^p\right]^{1/2}$$

(exponential f)

$$= r\left\{s_y^\infty - s_y^\circ \left[a + (1-a)\, e^{-r\zeta}\right]\right\} \left[d\underset{\sim}{e}^p : d\underset{\sim}{e}^p\right]^{1/2}$$

The form of the endochronic theory needed to produce the classical forms of plasticity is presented in Table 1.2.

Table 1.2 Comparison of The Classical Theories of Plasticity with the Endochronic Theories.

Classical Theories of Plasticity:

i) Isotropic Hardening(Prandlt and Reuss)

$H' = d\bar{\sigma}/d\bar{\xi}^p$ - rate of harding

$\bar{\sigma}$ - equivalent stress

$\bar{\xi}^p$ - eq. plastic strain

$$d\sigma_{kk} = (2\mu + 3\lambda)\, d\varepsilon_{kk}$$

$$d\underset{\sim}{S} = 2\mu\, d\underset{\sim}{\xi}' - \frac{\dot{\bar{\sigma}}\mu\,\underset{\sim}{S}\,(\underset{\sim}{S} : d\underset{\sim}{\xi}')\,\Gamma}{\bar{\sigma}^2\bar{\xi}^p\left[1 + (1/3\,\mu)H'\right]}$$

Endochronic Theory

$$\rho_1(z) = \underset{\sim}{h}^*(z) = \underset{\sim}{\alpha} = 0\ ;$$

$$f(\zeta) = 1 + (2/3)\,H'\zeta$$

ii) Linear Kinematic Harding (Prager [B25])

$$d\sigma_{kk} = (2\mu + 3\lambda)\, d\varepsilon_{kk}$$

$$d\underset{\sim}{S} = 2\mu\, d\underset{\sim}{\xi}' - \frac{\sigma\mu^2}{(C^* + 2\mu)(\sigma_y)^2}\,(\underset{\sim}{S} - \underset{\sim}{\alpha}) : d\underset{\sim}{\xi}'\,(\underset{\sim}{S} - \underset{\sim}{\alpha})$$

$$d\underset{\sim}{\alpha} = C^*\, d\underset{\sim}{\xi}'^p$$

Endochronic Theory

$$\rho(z) = \rho_{10} = C^*/2\mu \quad ; \quad \underset{\sim}{h}^* = 0 \quad ; \quad \underset{\sim}{S}^\circ_{ij} = \sqrt{\frac{2}{3}}\,\sigma^\circ_{ij}$$

$$f(\zeta) = 1.0$$

iii) Non-linear Kinematic Harding

(Mroz-Shrivastava-Dubey [B28])

(Eisenberg and Phillips [B27])

$$d\underset{\sim}{\alpha} = C(\zeta) \qquad d\underset{\sim}{\varepsilon}^{p} = D(\zeta)\,\underset{\sim}{\varepsilon}^{p}\,d\zeta$$

iv) Combined Isotropic and Kinematic Hardening

(Chaboche and Rousselier [B18])

Kinematic Harding:

$$d\underset{\sim}{\alpha} = \sum_{i} d\alpha^{(i)} \; ; \quad d\alpha^{(i)} = C_i\, d\underset{\sim}{\varepsilon}^{p} - D_i\,\alpha^{(i)}\,d\zeta$$

C_i and D_i are constants;

Isotropic Hardening:

$$S_y = Q\left(1 - e^{-b\zeta}\right) \; ; \quad dS_y/d\zeta = b\,(Q - S_y)$$

b, Q are constants

v) Perfect Plasticity

$$d\underset{\sim}{S} = 2\mu\, d\underset{\sim}{\varepsilon}' - \frac{(\underset{\sim}{S} - \underset{\sim}{\alpha})(\underset{\sim}{S} - \underset{\sim}{\alpha}) : d\underset{\sim}{\varepsilon}'}{c}$$

$$c = 1.0 \; ; \quad f(\zeta) = 1.0 \; ; \quad \rho(z) = 0$$

SELECTION OF POLYNOMIALS FOR INTERPOLATION OF THE STRESS

The presence of unwanted kinematic mechanism modes in the stress-based element is a primary concern when selecting the polynomial basis functions used to interpolate the stress field within an stress-based hybrid element. The kinematic mechanism modes that may arise due to a poor choice of stress polynomials are not unlike the mechanisms that may result when a displacement based element is subjected to reduced integration.

The criteria for the stability and convergence of discrete variational problems with Lagrange multipliers was the focus of the fundamental work of Ladyzhenskaya, Babuska and Brezzi (LBB) [B.28-30]. The LBB condition may be used as an a posteriori check of a formulation. While the work of LBB was limited to a variational statement with only one parameter, the multi-field case was the focus of the work of Xue and Atluri [B.31]. While the

satisfaction of the LBB condition will guarantee the convergence and stability of the formulation, it does not specify how the condition should be met. The work of Punch [B.32] and Punch and Atluri [B.33], addressed the problem of establishing criteria for the selection of stress polynomials such that the resulting element will be stable, invariant and least order.

In general, for a stress-based hybrid formulation, if the number of stress parameters β for an element is s, then the matrix $\underline{\underline{H}}$ should be a (sxs) positive definite symmetric matrix. The element stiffness matrix $\underline{\underline{K}} = \underline{\underline{G}}^T \underline{\underline{H}}^{-1} \underline{\underline{G}}$ should have a rank of (d-r) where d is the number of generalized nodal displacement and r is the number of rigid body modes. Thus, the matrix G, associating the assumed stress and displacement fields, is the most critical component of the formulation - the (sxd) homogeneous equation

$$\underline{\underline{G}}\ \underline{q} = \underline{0} \qquad (B.117)$$

should have, as its nontrivial solutions, only the r rigid body modes q . By virtue of the divergence theorem and the equilibrated stress field σ_{ij} , this expression can be written as

$$\underline{\beta}^T \underline{\underline{G}}\ \underline{q} = \int_S n_j\, \sigma_{ij}\, u_i\, dS = \int_V \sigma_{ij}\, \varepsilon_{ij}(u_k)\, dV \qquad (B.118)$$

where the following relation holds,

$$\int_{V_m} \sigma_{ij}\, \varepsilon_{ij}(\tilde{u}_k)\, dV \begin{cases} = 0 & \text{for rigid body mode} \\ > 0 & \text{for } (d-r) \text{ modes.} \end{cases} \qquad (B.119)$$

With $\varepsilon_{ij}(\tilde{u}_k)=0$ for r rigid modes q_r, the rank of G and consequently the overall rank of K, which it determines, is the minimum of (s,d-r) at best. For a formulation free of spurious energy modes, the minimum rank must be (d-r) and the number of chosen stress modes must therefore satisfy

$$s \geq d-r \qquad (B.120)$$

Noting that each extra term adds more stiffness [B.34], least-order selections (s = d-r) are considered to be best and are, of course, optimal with respect to compute resources.

The G matrix not only governs the existence, but is also central to the determination of convergence and stability through the LBB condition [B.29]. This convergence condition of functional analysis features G on a domain and states that, if there exists a $\beta > 0$ such that

$$\sup_{\forall \sigma_{ij} \in \bar{H}_1(\Omega_m)} \frac{\sum_{i=1}^{n} \int_{\Omega_m} \sigma_{ij}\, \varepsilon_{ij}(\tilde{u}_k)\, dV}{\| \sigma_{ij} \|_{H_1(\sigma)}} \geq \beta \| \tilde{u}_k \|_U \qquad (B.12)$$

$$\forall\ \tilde{u} \in U.$$

then the finite element problem has a unique solution. (B.119) and (B.121) are necessary and sufficient conditions for stability, respectively. When β is independent of mesh parameter h, convergence is then established. However, this theory only provides an posteriori check on a particular finite element formulation since the value and mesh dependence of β must be ascertained numerically in each case.

In addition to accomadating all reasonable load distributions, the chosen stress modes must be nonorthongal to the strain field in order to eliminate spurious zero energy modes and guarantee convergence. One possible approach to the eradication of mechanisms lies in the painstaking assembly of the G matrix from complete equilibrated polynomial stress, and strain tensors derived from the element displacement field. The rank of G may be computed by Gaussian elimination and stresses added or removed until the desired (d-r) value is reached. This rudimentary procedure, nevertheless, fails to address the requirement of coordinate invariance in the overall stress interpolation, as a result of which further criteria must be applied.

Coordinate invariance entails certain symmetry relations between the coordinates, relations which are governed by symmetry group theory. Although this theory applies exactly to perfect squares and cubes only, it nontheless provides a very effective approximation for distorted elements and generates a convenient sparse quasi-diagonal G matrix from which stress selections can easily be made. The mathematical foundation appears fully in [B.32], and the complete derivation for plane elements, as well as three-dimensional bricks, can be found in [B.32, B.33].

Considerable success has been achieved in approaches where the equilibrium constraints are relaxed on some [B.35] or all stress terms [B.36] by means of displacement field Lagrange multipliers. Taking advantage of the variation of natural coordinates in curvilinear elements, the stress tensor is expected as an unequilibrated summation of natural coordinate polynomials A_{ijm} with unknown β_m,

$$\sigma_{ij} = A_{ijm}\,\beta_m \qquad \text{in } V_n \tag{B.122}$$

Define

$$G_{ij} = \int_{V_m} A_{ijm}\,\psi_{ik,j}\,dV \tag{B.123}$$

where ψ_{ik} is interpolation functions for nodal displacement. With matrix G in this form, the derivation of stress modes for this hybrid stress formulation with a posteriori equilibrated local stress field can follow the procedures used by the formulation with a priori equilibrated stress stress field. The foregoing least-order stress polynomial selections in natural coordinate variable ξ_i are introduced into A_{ijm}, but this do not necessarily form stable, irreducible, invariant interpolations. However, it has been demonstrated in [B.32, B.33] that for the curvilinear elements if the stress mode is chosen to be of the same polynomial form as that of the stress mode which is derived by using group theory for squares and cubes, then the rank of $\underline{\underline{G}}$ is maintained to be (d-r) even for very severely distorted elements. Further, it has been clearly demonstrated [B.32, B.33] that the least-order,

invariant, isoparametric curvilinear hybrid elements are less distortion--sensitive and lead to more accurate results compared to the standard displacement elements in a variety of examples.

For the present 16-node isoparametric hybrid element, 42 stress component (= 48-6) should be chosen to form the least-order, stable, invariant element. The following stress polynomial selection was made based upon the suggestions provided in [86].

$$\Delta\sigma_{11} = \beta_1 + \beta_2 x + \beta_3 y + \beta_4 z + \beta_5 y^2 + \beta_6 xy + \beta_7 yz + \beta_8 xz + \beta_9 xyz + \beta_{10} y^2 z \qquad (B.124\text{-}c$$

$$\Delta\sigma_{22} = \beta_{11} + \beta_{12} x + \beta_{13} y + \beta_{14} z + \beta_{15} x^2 + \beta_{16} xy + \beta_{17} yz + \beta_{18} zx + \beta_{19} xyz + \beta_{20} x^2 z \qquad (B.124$$

$$\Delta\sigma_{33} = \beta_{21} + \beta_{22} x + \beta_{23} y - (\beta_{24} + \beta_{25}) z + \beta_{26} x^2 + \beta_{27} y^2 + \beta_{28} xy + \beta_{29} x^2 y + \beta_{30} xy^2 \qquad (B.124$$

$$\Delta\sigma_{12} = \beta_{31} + \beta_{32} x + \beta_{33} y + \beta_{34} z - \beta_8 yz - \beta_{17} xz - \frac{1}{2}\beta_{16} x^2 - \frac{1}{2}\beta_6 - \frac{1}{2}\beta_{19} x^2 z - \frac{1}{2}\beta_9 y^2 z \qquad (B.12$$

$$\Delta\sigma_{13} = \beta_{35} + \beta_{24} x + \beta_{36} y - (\beta_2 + \beta_{33}) z + \beta_{39} xy - \frac{1}{2}\beta_{37} x^2 + \beta_{38} \qquad (B.1$$

$$\Delta\sigma_{23} = \beta_{40} + \beta_{25} x + \beta_{41} y - (\beta_{13} + \beta_{32}) z + \beta_{37} xy + \beta_{42} x^2 - \frac{1}{2}\beta_{39} y^2 \qquad (B.12$$

INTERPOLATION FUNCTIONS FOR DISPLACEMENT

Standard isoparametric shape functions were chosen for this 16-node stress-based hybrid element. The local coordinates and nodal numbering are shown in Fig. B1, and the 16 shape functions are presented as following:

$$N_1 = \frac{1}{8}(1-\xi)(1-\eta)(1-\zeta) - \frac{1}{2}N_9 - \frac{1}{2}N_{12}$$
$$N_2 = \frac{1}{8}(1+\xi)(1-\eta)(1-\zeta) - \frac{1}{2}N_9 - \frac{1}{2}N_{10}$$
$$N_3 = \frac{1}{8}(1+\xi)(1+\eta)(1-\zeta) - \frac{1}{2}N_{10} - \frac{1}{2}N_{11}$$
$$N_4 = \frac{1}{8}(1-\xi)(1+\eta)(1-\zeta) - \frac{1}{2}N_{11} - \frac{1}{2}N_{12}$$
$$N_5 = \frac{1}{8}(1-\xi)(1-\eta)(1+\zeta) - \frac{1}{2}N_{13} - \frac{1}{2}N_{16}$$
$$N_6 = \frac{1}{8}(1+\xi)(1-\eta)(1+\zeta) - \frac{1}{2}N_{13} - \frac{1}{2}N_{14}$$
$$N_7 = \frac{1}{8}(1+\xi)(1+\eta)(1+\zeta) - \frac{1}{2}N_{14} - \frac{1}{2}N_{15}$$
$$N_8 = \frac{1}{8}(1-\xi)(1+\eta)(1+\zeta) - \frac{1}{2}N_{15} - \frac{1}{2}N_{16}$$
$$N_9 = \frac{1}{4}(1+\xi)(1-\xi)(1-\eta)(1-\zeta)$$
$$N_{10} = \frac{1}{4}(1+\eta)(1+\xi)(1-\eta)(1-\zeta)$$
$$N_{11} = \frac{1}{4}(1-\xi)(1+\xi)(1+\eta)(1-\zeta)$$
$$N_{12} = \frac{1}{4}(1-\eta)(1-\xi)(1+\eta)(1-\zeta)$$
$$N_{13} = \frac{1}{4}(1+\xi)(1-\xi)(1-\eta)(1+\zeta)$$
$$N_{14} = \frac{1}{4}(1+\eta)(1+\xi)(1-\eta)(1+\zeta)$$
$$N_{15} = \frac{1}{4}(1-\xi)(1+\xi)(1+\eta)(1+\zeta)$$
$$N_{16} = \frac{1}{4}(1-\eta)(1-\xi)(1+\eta)(1+\zeta)$$

(B.125)

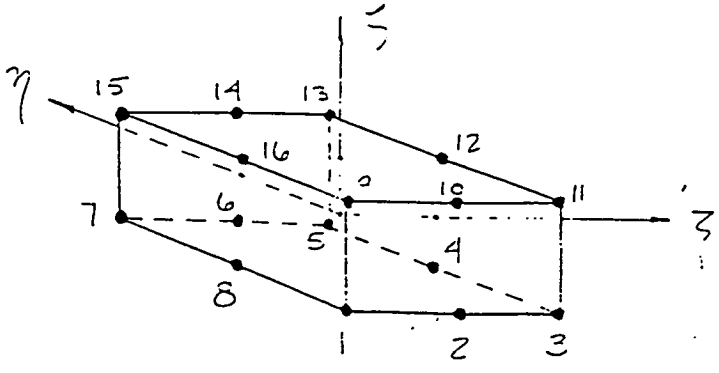
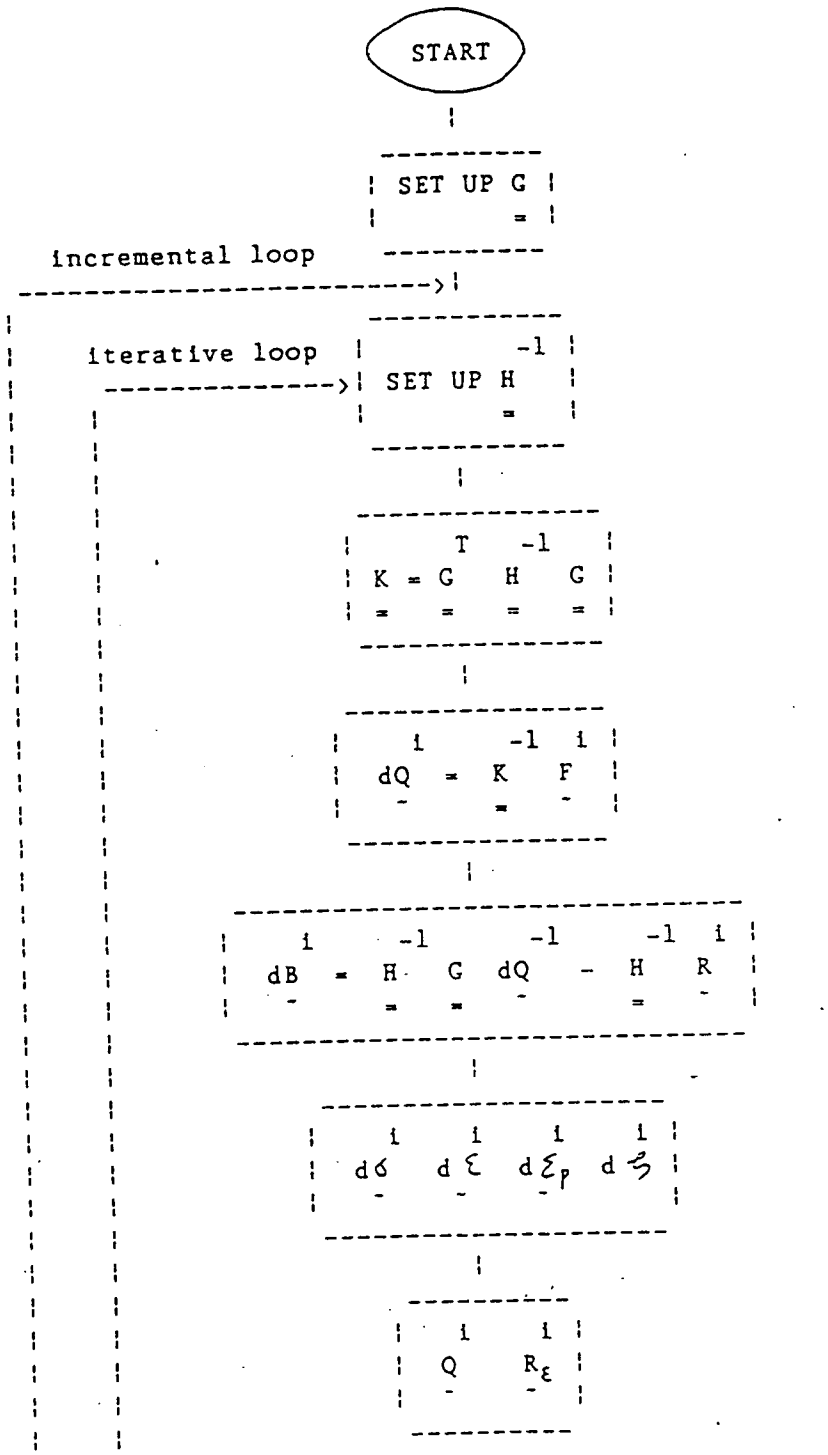
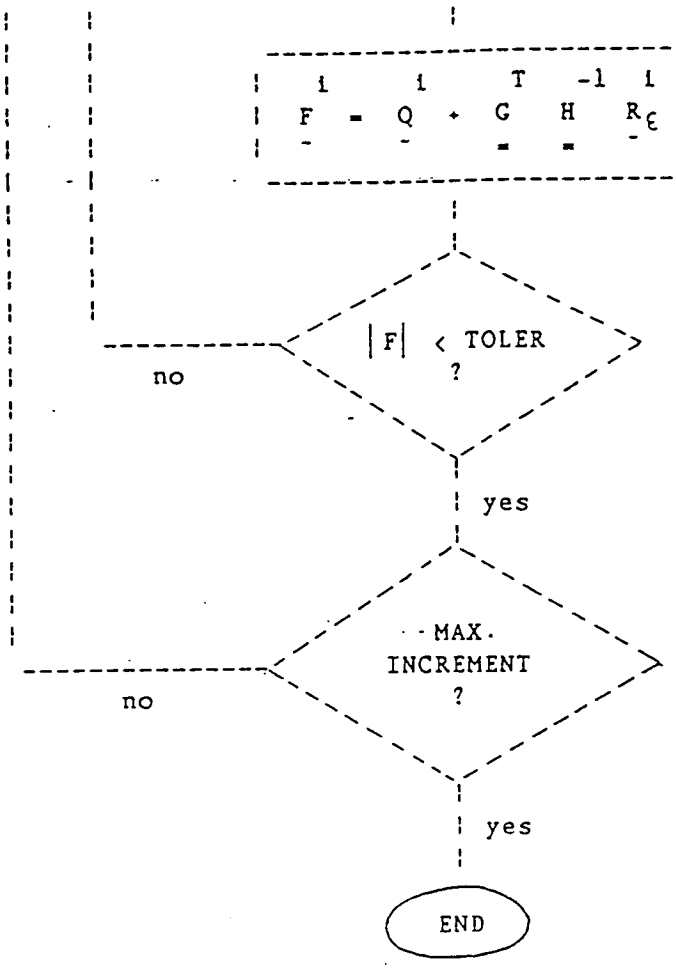


Fig. B1. Definition of element nodal numbering and natural coordinate.

FLOW CHART FOR THE IMPLEMENTATION OF 16-NODE STRESS-BASED HYBRID ELEMENT




$F^{i} = Q^{i} + G^{T} H^{-1} R_{\epsilon}^{i}$
$|F| < TOLER$?
no
yes
MAX. INCREMENT ?
no
yes
END

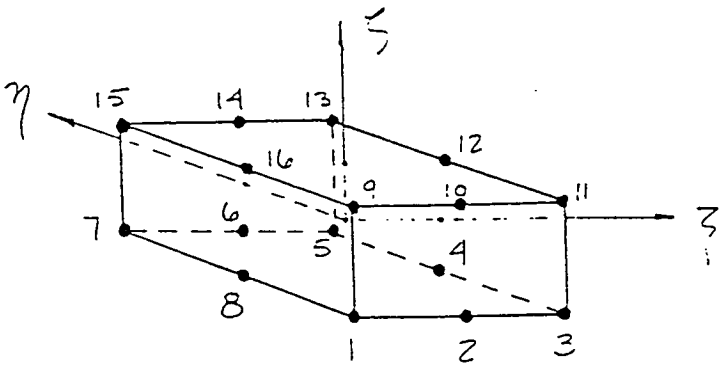
USER MANUAL

There are two keywords and some modifications in the parameter data section of NESSUS input data file.

* ELEMENTS

There is one more element type

50 : 16-node hybrid shell element. None of the five element properties are effected.



* HYBR

This option enables the hybrid shell element. One parameter is required .

parameter 1 : "42" , NSTSFN , number of stress function parameters.

* ENDO

This keyword invokes the endochronic theory. no parameter is required.

NOTE : When * HYBR is flagged , * BFGS can not be used.

There are two new keywords in the model data section of NESSUS input data file.

* HYPR

This data segment is used to specify the material properties of the hybrid shell element. Five real numbers are required. These are (1) thickness (dummy (2) Young's modulus, (3) Poisson's ratio, (4) initial yield stress, and (5) strain hardening coefficient.

If * ENDO is flagged, the last two property data are ignored.

* ENDO

This data segment is used to specify the endochronic theory properties of the hybrid shell element. A maximum of ten property data can be specified.

In the linear strain hardening case, the first data is $\rho_o = \frac{S_y}{2\mu} = \frac{\bar{\sigma}}{\sqrt{\epsilon}\mu}$, the second data is $H'_e = \frac{1}{S_y^o}\frac{d\bar{\sigma}}{d\bar{\epsilon}}$, and the others are dummy.

* PROP

If * HYPR is flagged, * PROP is ignored.

SUMMARY OF TAPES

There are eight new data files in the hybrid shell element portion of NESSU!

(1) HYBR.DAT (Tape 3)

This files contains $\underset{\approx}{H}^{-1}$ and $\underset{\approx}{G}$ for each element. $\underset{\approx}{G}$ is calculated at the begining. $\underset{\approx}{H}^{-1}$ is calculated at each iteration.

(2) STRES.DAT (Tape 2)

This files contains the stress vector of each Gaussian point.

(3) RES.DAT (Tape 7)

This file contains residual forces arising from enforcing the compatibility condition.

$$R_k^{\varepsilon i} = \int_V A_{mnk} \left\{ \varepsilon_{mn}^{i} - \frac{1}{2} \left(U_{m,n} - U_{n,m} \right)^{i} \right\} dV$$

These residual forces are used to calculate , the stress parameter increment.

(4) EPIND.DAT (Tape 10)

This file contains an index of each Gaussian point indicating whether it is elastic or plastic.

(5) STRAN.DAT (Tape 13)

This file contains the strain vector of each Gaussian point . These strain vector are calculated from $\underset{\approx}{\varepsilon}$-$\underset{\sim}{\sigma}$ relations.

IMPORTANT SUBROUTINES AND VARIABLES

1) CAZETA

This subroutine is to calculate the incremental internal time parameter

DEPL	: incremental plastic strain
DZETA	: incremental internal time parameter

2) D3D16N

This subroutine is to calculate the shape function derivative for the 16-nc shell element. All the variables are the same as the other similiar subroutir

3) FORMAM

This subroutine is to set up the A matrix in each Gaussian point.

G, H, and Q	: natural coordinate of the gaussian points
D	: A matrix
XINVER	: base vectors of the centrial curvilinear coordinate
NSTSFN	: number of stress function been used, 42

4) FORMBV

This subroutine is to calculate incremental stress parameter

BETAIN	: incremental stress parameter
DISWRK	: displacement increment
ELEM1	: H matrix
ELEM2	: G matrix
RSTRAN	: residual force arised from compatibility

4) FORMCM

This subroutine is to set up the elastic strain-stress relation matrix, C, at each Gaussian point.

CMATRX	: C matrix
CHAR	: elastic material property at Gaussian points

5) FORMGM

This subroutine is to set up the G matrix. It is called only once for each element.

GMATRX	: G matrix

(6) EPSN.DAT (Tape 22)

This file contains the total plastic strain vector and the incremental plastic strain vector of each Gaussian point.

(7) ZETA.DAT (Tape 14)

This file contains the total internal time parameter ζ and the incremental internal time parameter $\delta\zeta$ of each Gaussian point.

(8) VALGLO.DAT (Tape 30)

This file contains the accumulated nodal values of $\tilde{\sigma}$, $\tilde{\varepsilon}$, $\tilde{\varepsilon}^p$, and ζ. These values must be divided by the number of elements which contain the node to get the average values at the node.

This subroutine is to set up the plastic stress-strain relation matrix, D

DMATRX	: D matrix
HYSIG	: stress vector
ENCHAR	: endochronic theory property
STEMP	: stress deviator

13) PLASTC

This subroutine is to set up the plastic strain-stress relation matrix, C.

CMATRX	: C matrix

14) RESID

This subroutine is to set up the residual forces arised from equilibrium a compatibility.

DISTOT	: total displacement
DISINC	: total incremental displacement up to iteraton i
RXII	: incremental displcaement of iteration i
XP	: equilibrium residual force , total residual force later
XP2	: compatibility residual force
STRN1	: strain (strain-displacement)
STRN2	: strain (strain-stress)

15) S3D16N

This subroutine is to set up the shape functions of 16-node shell element.A the variables are the same as the other similiar subroutines.

16) SIGBAR

This subroutine is to calculate the effective stress and stress deviator.

SIG	: stress vector
S	: stress deviator
SEQ	: effective stress

17) UPEPSN

This subroutine is to update the total plastic strain and total internal tim parameter after each load increment.

18) YIEL2

This subroutine is to calculate the radius of yield surface from internal

COOR : coordinate of each nodal point in one element

6) FORMHM

This subroutine is to set up the H matrix. It is called in each iteration for each element.

HMATRX : H matrix
EPIND : index of elastic/plastic for each Gaussian point

7) GHOOK

This subroutine is to set up the elastic stress-strain relation matrix, D, at each Gaussian point.

DMATRX : D matrix

8) HYOUT

This subroutine print out the displacement of the hybrid shell element.

9) HYSTIF

This subroutine is to set up the element stiffness matrix for the hybrid shell element.

EHSTIF : element stiffness matrix
ELMRHS : element force vector ; it is set to zero now

10) HYSTSS

This subroutine is to calculate the stress, strain, and plastic strain.

EPSN : total plastic strain
SIG : stress in iteration i-1
SIGTT : stress in iteration i
STRN2 : strain came form strain-stress relations
DDSTRN : incremental strain
ZETA : total interal time parameter
DEPSN : incremental plastic strain
DZETA : incremental interal time parameter

11) LSNODE

This subroutine is to calculate nodal values of stress, strain, plastic strain and internal time parameter.

12) PLADMT

time parameter ZETA.

SYT	: radius of yield surface
FP	: strain hardening coefficient

SUMMARY OF LOGIC FOR SUBROUTINE HYSTSS

The Gaussian point had previously yielded. Now check to see if $\bar{\sigma}_y^i > \bar{\sigma}_y^{i-1}(\zeta)$ where $\bar{\sigma}_y^i$ is the effective stress of iteration i, $\bar{\sigma}_y^{i-1}(\zeta)$ is the radius of yield surface of iteration i-1.

NO

The Gaussian point is unloading elastically. Calculate $\Delta\underset{\sim}{\xi}^i = \underset{\approx}{C}\,\Delta\underset{\sim}{\sigma}^i$ and return

YES

The Gaussian point had yielded previously and the stress is sti increasing. Calculate $\Delta\underset{\sim}{\xi}_p^i$, $\Delta\underset{\sim}{\xi}^i$. Where

$$\Delta\underset{\sim}{\xi}_p^i = \frac{\Delta\underset{\sim}{\sigma}^i : \underset{\sim}{n}_m}{\underset{\sim}{n}_m : \underset{\approx}{C} : \underset{\sim}{n}_m \; \bar{\sigma}_y' \; f'(\bar{\varepsilon}_p)}$$

$$\underset{\sim}{S}_m = \underset{\sim}{S}^{i-1} + \frac{1}{2}\Delta\underset{\sim}{\sigma}^i$$

$$\underset{\sim}{n}_m = \frac{\underset{\sim}{S}_m}{\bar{\sigma}_m} : \frac{\underset{\sim}{S}_m}{\bar{\sigma}_m}$$

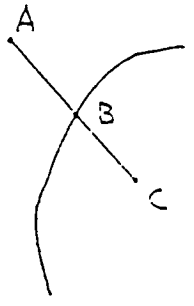
The Gaussian point had not previously yielded.
Now check to see if $\bar{\sigma}_y^i > \bar{\sigma}_y^{i-1}$ (ζ)

NO

The Gaussian point is still elastic. Calculate $\Delta \underset{\sim}{\xi}^i = \underset{\sim}{\underset{\sim}{C}} \Delta \underset{\sim}{\sigma}^i$ and return

YES

The Gaussian point has yielded during the iteration. The portion of the stress greater than the yield value must be reduced to the yield surface. The reduction factor R is given from fig. below to be R = AB/AC . Then use conventional displcement based plastic scheme to calculate the overshot portion of $\Delta \underset{\sim}{\sigma}^i$ and corresponding $\Delta \underset{\sim}{\xi}^i$.



REFERENCES

[B.1] Pian, T.H.H., and Tong, P., 'Basis of Finite Element Methods for Solid Continua,' Int. J. Num. Meth. Engng., 1, 1969, pp. 3-28.

[B.2] Yamada, Y., Nakagiri, S. and Takatsuka, K., 'Elastic-Plastic Analysis of St. Venant Torsion Problem by a Hybrid Stress Methods,' Int. J. Num. Meth. Engng., 5, 1972, pp. 193-207.

[B.3] Luk, C.H. 'Assumed Stress Hybrid Finite-Element Method for Fracture Mechanics and Elasto-Plastic Analysis,' MIT ASRL TR 170-1,1972.

[B.4] Nyssen, C. and Beckers, P., 'A Unified Approach for Displacement, Equilibrium and Hybrid Finite Element Models in Elasto-Plasticity,' Comp. Meth. in Appl. Mech. Engng., 44, 1984, pp. 131-151.

[B.5] Mroz, Z., 'On the Description of Anisotropic Workhardening,' J. Mech. Phys. Sol., Vol. 15, pp. 163, 1967.

[B.6] Mroz, Z., 'An Attempt to Describe the Behavior of Metals Under Cyclic Loads Using a More General Workhardening Model,' Acta Mech., Vol. 7, pp. 199, 1969.

[B.7] Dafalias, Y.F., and Popov, E.P., 'A Model of Nonlinearly Hardening Materials for Complex Loading,' Acta Mech., Vol. 21, pp. 173, 1975.

[B.8] Dafalias, Y.F., and Popov, E.P., 'Plastic Interal Variables Formalism of Cyclic and Anisotropic,' J. Appl. Mech., Vol. 43, pp. 645, 1976.

[B.9] Krieg, R.D., 'A Pratical Two Surface Plasticity Theory,' J. Appl. Mech., Vol. 42, pp. 641, 1975.

[B.10] Valanis, K.C., 'Fundamental Consequence of a New Intrinsic Time Measure-Plasticity as a Limit of the Endochronic Theory,' Arch. Mech., Vol. 32, pp. 171, 1980.

[B.11] Watanabe, O., and Atluri, S. N., 'Constitutive Modeling of Cyclic Plasticity and Creep, Using an Internal Time Concept,' Technical Report, Center for the Advancement of Computational Mechanics, Georgia Tech., 1984.

[B.12] Onat, E.T., 'Representation of Inelastic Mechanical Behavior by Means of State Variables,' IUTAM Symp. Thermoinelasticity, East Kilbride (Ed. Boly, B.A.), Springer-Verlag, 213, 1968.

[B.13] Onat, E.T., 'The Notion of State and Its Implications in Thermodynamic of Inelastic Solids,' IUTAM Symp. Irreversible Aspects of Continuum Mechanics and Transfer of Physical Characteristics in Moving Fluids, Vienna (Ed. Parkus, H. and Sedov, L.I.),Springer-Verlag, pp. 292, 1966.

[B.14] Onat, E.T., 'Representation of Inelastic Behavior in the Presents of Anisotropy and Finite Deformations,' Proc. of Res. Workshop on Plasticity of Metals at Finite Strain: Theory, Experiment and Computation, Stanford Univ., (Ed. Lee, E.H., and Mallott, R.L.), pp. 519, 1981.

[B.15] Fardshisheh, F. and Onat, E.T., 'Representation of Elastoplastic Behavior by Means of State Variables,' Int. Symp. Foundations of Plasticity, 2, Problems of Plasticity (Ed. Sawczuk, A.), Noordhoff Publishers, pp. 89, 1973.

[B.16] Onat, E.T. and Fardshisheh, F., 'Representation of Creep, Rate Sensitivity and Plasticity,' SIAM J. Appl. Math., Vol. 25, pp. 522, 1973.

[B.17] Chaboche, J.L., 'Viscoplastic Constitutive Equations for the Description of Cyclic and Anisotropic Behavior of Metals,' Bull. L'Aca. Polo. Sci., Ser. Sci. Tech., XXV, pp. 33, 1977.

[B.18] Chaboche, J.L. and Rousselier, G., 'On the Plastic and Viscoplastic Constitutive Equations, Part I: Rules Developed with Internal Variables Concept; Part II:Application of Internal Variables Concept to the 316 Stainless Steel,' Inelastic Analysis and Life Prediction in Elevated Temperature Design, ASME PVY Vol. 59, pp 33, 1982.

[B.19] Watanabe, O. and Atluri, S.N., 'Internal Time, General Internal Variable, and Multi-Yield-Surface Theories of Plasticity aand Creep: A Unification of Concepts,' Int. J. Plasticity, Vol. 2, pp.37-57,1986.

[B.20] Valanis, K.C., 'A Theory of Viscoplasticity without a Yield Surface, Part I. General Theory,' Arch. Mech., Vol. 23, No. 4, pp. 517-533, 1971.

[B.21] Valanis, K.C., 'A Theory of Viscoplasticity without a Yield Surface, Part II. Application to Mechanical Behavior of Metals,'Arch. Mech., Vol. 23, No. 4, pp. 535-551,1971.

[B.22] Rivlin, R.S., 'Some Comments on the Endochronic Theory of Plasticity,' Int. J. of Solids & Struc., Vol. 17, pp. 232-248, 1981.

[B.23] Valanis, K.C., 'Fundamental Consequence of a New Intrinsic Time Measure-Plasticity as a Limit of the Endochronic Theory,' Arch. Mech., Vol. 32, No. 2, pp. 171-191, 1980.

[B.24] Valanis, K.C. and Fan, J., 'Endochronic Analysis of Cyclic Elastoplastic Strain Fields in a Notched Plate,' J. of Appl. Mech., Trans. ASME, Vol. 50, No. 4a, pp. 789-794, 1983.

[B.25] Praegr, W., 'A New Method of Analyzing Stress and Strain in Work Hardening Plastics Solids,' J. of Appl. Mech., Vol. 23, pp. 493-496, 1956.

[B.26] Mroz, Z., Shirivastava, H.P., and Dubey, R.N., 'A Non-Linear Hardening Model and its Applications to Cyclic Loading,' ACTA Mechanica, Vol. 25, pp. 51-61, 1976.

[B.27] Eisenberg, M.A. and Phillips, A. 'On Non-Linear Kinematic Hardening,' ACTA Mechanica, Vol. 5, pp. 1-13, 1968.

[B.28] Brezzi, F., 'On the Existence, Uniqueness and Approximation of Saddle-Point Problems Arising from Lagrange Multipliers,' RAIRO 8-R2, pp. 129-151, 1974.

[B.29] Babska, I., Oden, J.T., and Lee, J.K., 'Mixed-Hybrid Finite Element Approximation of Second Order Elliptic Boundary Value Problems, Part I,' Comp. Meth. Appl. Mech. Engng., 11, pp. 175-205, 1977.

[B.30] Babska, I., Oden, J.T., and Lee, J.K., 'Mixed-Hybrid Finite Element Approximation of Second Order Elliptic Boundary Value Problems, Part II, Weak Hybrid Methods,' Comp. Meth. Appl. Mech. Engng., 14, pp. 1-22, 1978.

[B.31] Xue, W-M., Karlovitz, L.A. and Atluri, S.N., 'On the Existence and Stability Conditions for Mixed-Hybrid Finite Element Solutions Based on Reissner's Variational Principle,' Int. J. Solids & Struc., VOl. 21 No. 1, pp. 97-116, 1985.

[B.32] Punch. E.F., Ph. D. Thesis, 'Stable, Invariant, Least-Order Isoparametr Mixed-Hybrid Stress Elements: Linear Elastic Continua, and Finitely Deformed Plates and Shells,' School of Civil Engineering, Georgia Tech. August, 1983.

[B.33] Punch, E.F., and Atluri, S.N., 'Development and Testing of Stable, Invariant, Isoparametric Curvilinear 2- and 3-D Hybrid-Stress Elements,' Comp. Meth. in Appl. Mech. Engrg., Vol. 47, pp. 331-356, 1984.

[B.34] Henshell, R.D., 'On Hybrid Finite Elements', in Mathematics of Finite Elements and Applications, (Ed. Whiteman, J.R.), Academic Press, 1973.

APPENDIX C

Monte Carlo Programs for Probabilistic Structural Analysis

Mr. Seung J. Lee
Mr. Torng Yi
Prof.Paul H. Wirsching

University of Arizona

1.0 INTRODUCTION

1.1 Introductory Remarks

Monte Carlo traditionally has been considered to be a "last resort" method for solving a probability or statistics problem because of high cost relative to accuracy of the results. However, in recent times a combination of the development of new efficient numerical techniques and new digital computing hardware have made Monte Carlo more attractive.

Presented in this report are descriptions of the following Monte Carlo programs dedicated to probabilistic structural analysis.

1. "Conventional" Monte Carlo
2. Variance reduction using antithetic variates
3. Direct evaluation of the probability integral
4. The Harbitz method

Provided in the following sections are descriptions of how each method works as well as a comprehensive study of the performance of each.

1.2 The Basic Problems

Consider the random variable Z as a function of the random vector

$$\underset{\sim}{X} = (X_1, X_2, \ldots X_n)$$

$$Z = h(\underset{\sim}{X}) \tag{1.1}$$

The distribution of each X_i is known. It is assumed that all X_i are mutually independent.

One problem of probabilistic mechanics and design is to compute a point probability,

$$p = P[h(\underset{\sim}{X}) \leq h_o] \tag{1.2}$$

For example, p could represent the probability of exceedance of a deflection or perhaps the probability of failure.

The second problem is the extension of the first to the construction of a cumulative distribution function.

$$F_Z(z) = P[h(\underset{\sim}{X}) \leq z] \tag{1.3}$$

Clearly the two problems are identical, but optimal strategies for analysis may differ. For example, to construct the CDF, one option would be to obtain point estimates of F_Z at selected values of z, then fit a curve through the points. A second option would be to construct an empirical distribution function from a large sample of Z_i (See Sec. 2.4).

1.3 Random Samples

The basis for Monte Carlo simulation is a standard uniform distribution random number generator. Methods of generating uniform variates are generally based on recursive calculations of residues of modulus m from a linear transformation [1]. Most large computers have such a generator as a library function.

A variety of methods can be employed to generate variates from the distributions. Presented in Appendix A are algorithms used for the program presented herein.

2.0 "CONVENTIONAL" MONTE CARLO

2.1 Point Probability Estimates by Conventional Monte Carlo Using the Bernoulli Parameter

Consider a function, $h(\underset{\sim}{X})$, where $\underset{\sim}{X}$ is a vector of random variables, all having known distributions. It is required to compute,

$$p = P[h(\underset{\sim}{X}) \leq h_o] \tag{2.1}$$

The problem can be reformulated as

$$p = P[g(\underset{\sim}{X}) \leq 0] \tag{2.2}$$

where $g(\underset{\sim}{X})$, called the "performance function," is

$$g(\underset{\sim}{X}) = h(\underset{\sim}{X}) - h_o \tag{2.3}$$

In a direct Monte Carlo scheme, a sequence of K random vectors, $\underset{\sim}{X}_i$, can be sampled, and in turn, a sequence of g_i; i = 1, K computed. Define

$$Y_i = \begin{cases} 1 & \text{if } g_i \leq 0 \\ 0 & \text{if } g_i > 0 \end{cases} \tag{2.4}$$

Thus, Y_i has a Bernoulli distribution

$$P(Y_i = 1) = p \tag{2.5}$$

$$P(Y_i = 0) = 1 - p$$

where the Bernoulli parameter p is the same p as in Eq. 2.1.

The maximum likelihood estimate (MLE) of p is [5],

$$\hat{p} = \frac{1}{K} \sum_{i=1}^{K} Y_i \tag{2.6}$$

But $\sum Y_i$ is just the total number of $g_i \leq 0$, denoted as N_o. Thus, $\hat{p}$ is just the fraction of the g_i's less than zero

$$\hat{p} = \frac{N_o}{N} \tag{2.7}$$

A flow diagram of conventional Monte Carlo is given in Fig. 2.1. A listing of a computer program for conventional Monte Carlo employing the Bernoulli parameter is provided in Appendix B and an example of the output is shown in Fig. 2.2.

2.2 Confidence Intervals on the Bernoulli Parameter, p

The MLE of p is $\hat{p}$. Because of sampling error, $\hat{p}$ is only an estimate, and the key question is how close is p to $\hat{p}$. Confidence intervals are described below. Note that these confidence intervals refer to sampling error of the Monte Carlo process, not uncertainties associated with the parameters of X_i.

Consider $\hat{p}$,

$$\hat{p} = \frac{1}{K} \sum_{i=1}^{K} Y_i \tag{2.8}$$

The mean and variance of $\hat{p}$ are [5]

$$E(\hat{p}) = p \tag{2.9}$$

$$V(\hat{p}) = \frac{p(1 - p)}{K} \tag{2.10}$$

By the central limit theorem, $\hat{p}$ will approach a normal distribution as $K \to \infty$. Confidence intervals for p are constructed using normal distribution mathematics,

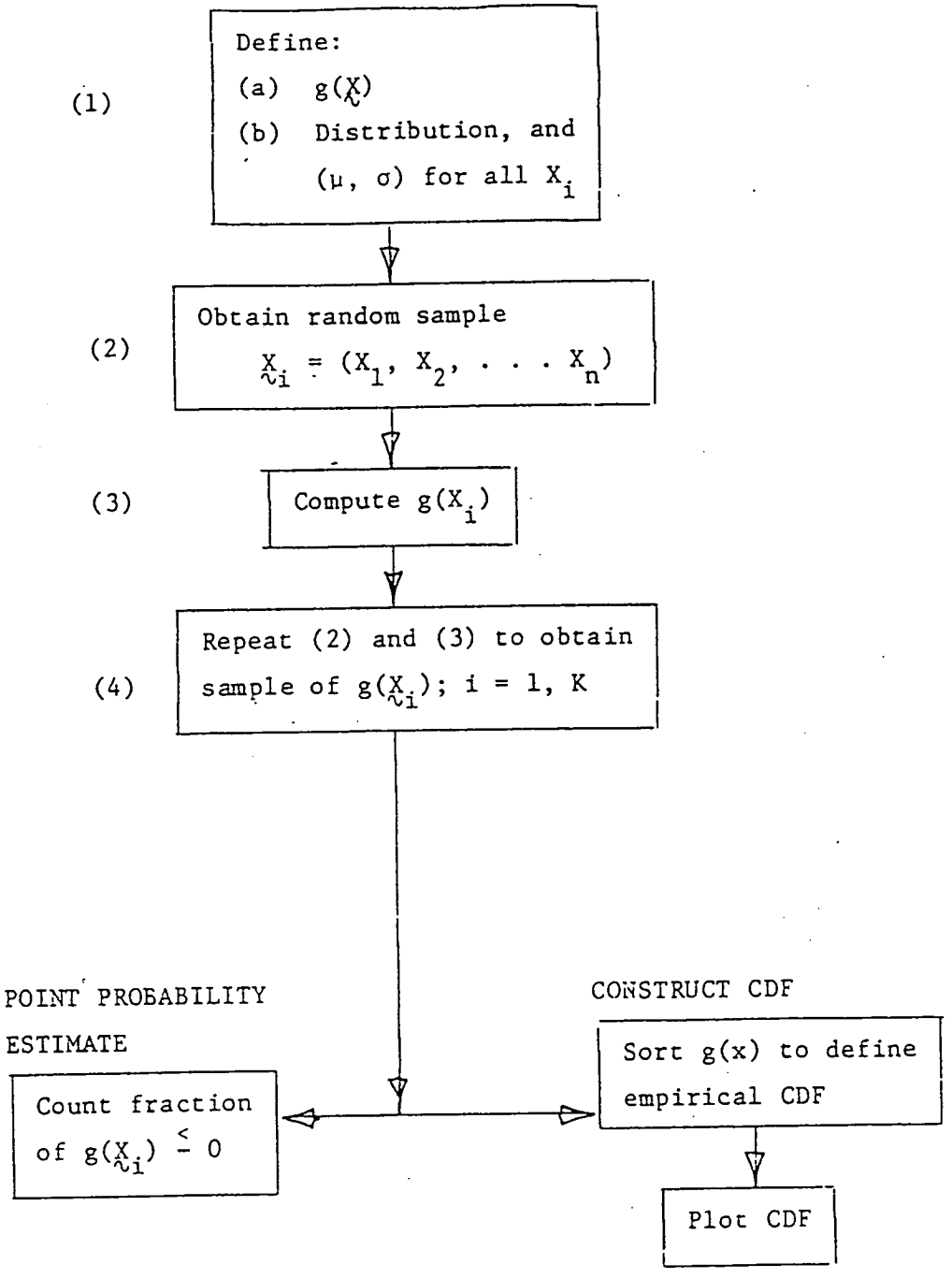


Fig. 2.1 Flow diagram of conventional Monte Carlo

MONTE CARLO SOLUTION

LIMIT STATE FUNCTION : R=S

SAMPLE SIZE, K= 100

NUMBER OF RANDOM VARIABLES, N= 2

RANDOM VARIABLES

VARIABLE	DISTRIBUTION	MEAN	STD DEV
R	WEIBULL	.20000E+02	.20000E+01
S	EVD	.10000E+02	.20000E+01

STATISTICS OF Y : ← Note that Y is the same as g(X); these are the statistics on the limit state function.

MEAN = .10018E+02

STD DEV = .27499E+01

MEDIAN = .96606E+01

COV = .27450E+00

This is $\hat{p}$

NUMBER OF NEG Y VALUES= 0. PERCENT OF TRIALS= .000000

Fig. 2.2 Output of conventional Monte Carlo program. (No sorting requested)

Performance function; g(R,S) = R - S

$$\hat{p} - z_{\alpha/2}\sqrt{\frac{\hat{p}(1-\hat{p})}{K}} \leq p \leq \hat{p} + z_{\alpha/2}\sqrt{\frac{\hat{p}(1-\hat{p})}{K}} \qquad (2.11)$$

where $\hat{p}$ is substituted for p in the variance. The probability that p will be bounded by the lower and uppper limit is $1-\alpha$, where α is the confidence coefficient. $z_{\alpha/2}$ is the standard normal variate corresponding to $\alpha/2$. Commonly used values

α	$z_{\alpha/2}$
.10	1.64
.05	1.96
.01	2.58

The confidence interval of Eq. 2.11 relies on the central limit theorem and must be considered as only an approximation for finite K. In general, the approximation is considered "valid" if $Kp > 5$ [5].

Eq. 2.11 can be written as,

$$\hat{p}\,(1-\gamma) \leq p \leq \hat{p}(1+\gamma) \qquad (2.12)$$

where,

$$\gamma = \frac{z_{\alpha/2}}{\hat{p}}\sqrt{\frac{\hat{p}(1-\hat{p})}{K}} \qquad (2.13)$$

Eq. 2.13 is displayed in Figs. 2.3 and 2.4 for 90% and 95% confidence intervals respectively. These figures show the sample size requirements for confidence intervals of a given width and level. For example, if the point probability is expected to be about 10^{-3}, and it is required to have p within $\pm$ 10% of p with a confidence of 90%, then it is necessary to have a sample of size $K > 200{,}000$.

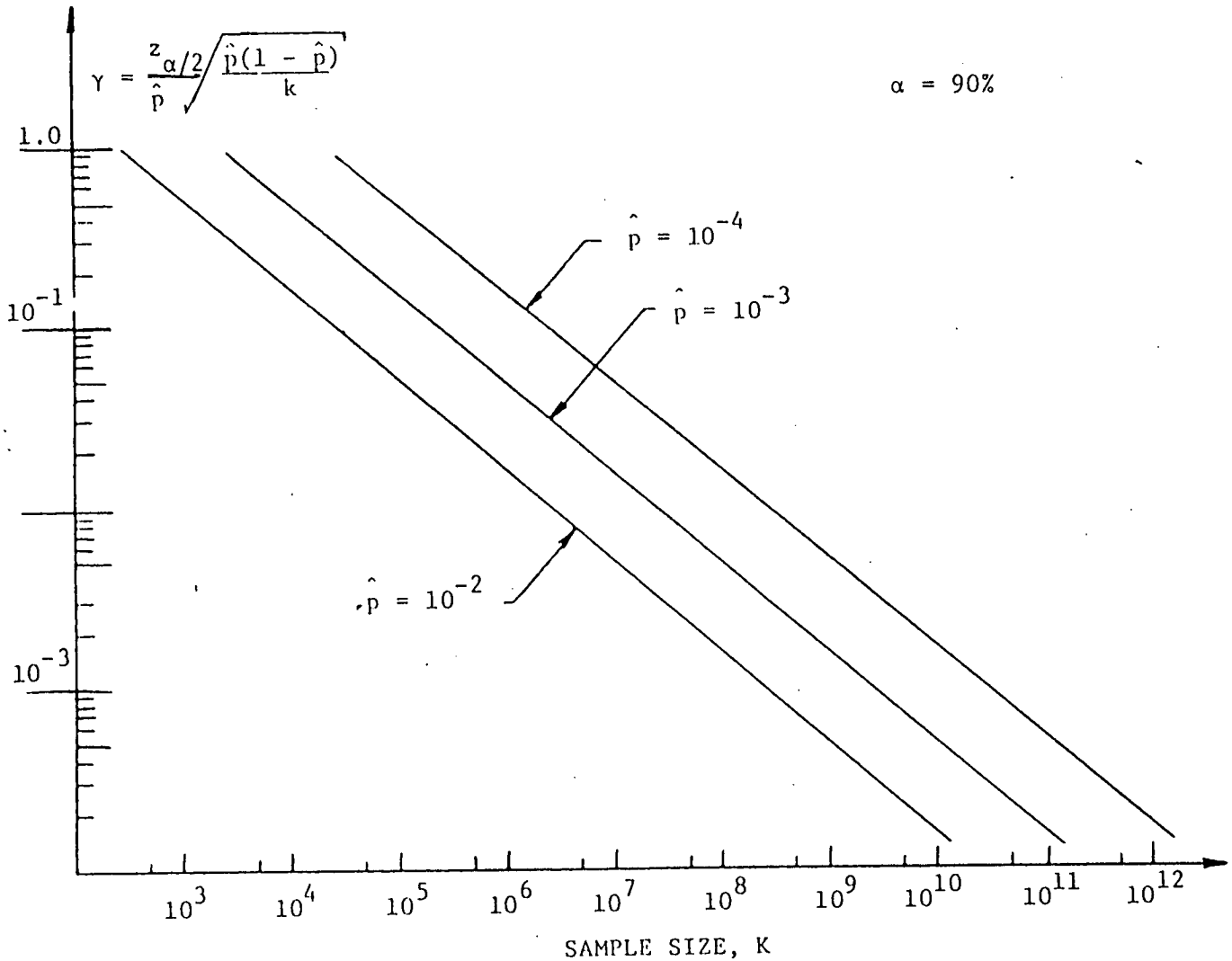


Fig. 2.3 90% confidence intervals on p as a function of sample size and $\hat{p}$

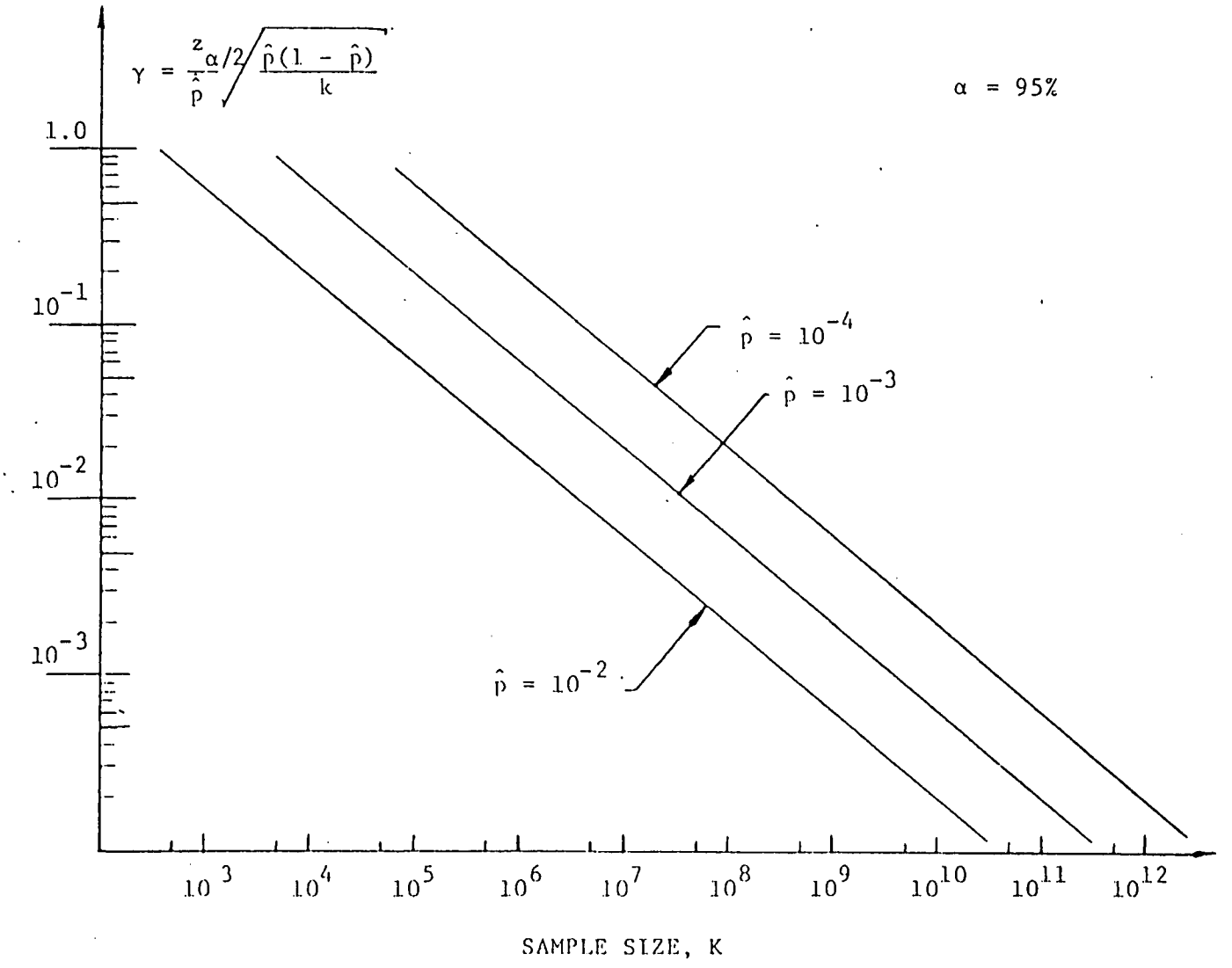


Fig. 2.4 95% confidence intervals on p as a function of sample size and $\hat{p}$.

2.3 Computer CPU Time on the CYBER 175

The conventional Monte Carlo program of Appendix B was exercised on several problems using all five of the available distributions. CPU time was recorded for each program. It is assumed that this conventional Monte Carlo program will provide an upper bound to CPU time relative to other, and more efficient, Monte Carlo schemes. The CYBER 175 is the mainframe computer at the University of Arizona, and all results relate to this machine.

Recorded CPU time for several examples was consistent. Compilation and loading time for all cases are shown in Table 2.1. These are average values, but there was little variation.

Execution CPU time essentially depends only upon the number of variables and not on distributional forms or performance functions. Fig. 2.5 illustrate the CPU execution time per variate as a function of sample size K. Total CPU time is obtained by adding compilation and loading time to execution time.

A sample program was run on both the CYBER 175 and the VAX 11/780 for a time comparison. The results shown in Table 2.2, reaffirm the fact that the VAX is too slow for production Monte Carlo.

To get an idea of computer charges for running Monte Carlo, Fig. 2.6 is provided. This is the commercial rate of the UA CYBER 175 for low priority jobs.

Table 2.1

Compilation and Loading CPU Time for Conventional Monte Carlo Program on CYBER 175

	CPU Time (sec)
Compile	1.0
Load	0.25

Table 2.2

Comparison of CPU time Between CYBER 175 and VAX 11/780 for one Example Problem*

	Time (sec)	
	CYBER 175	VAX 11/780
Compile	1.0	14
Link	0.25	5
Execution	7.5	30
TOTAL	8.75	49.0

*There were 2 variables; K = 30,000.

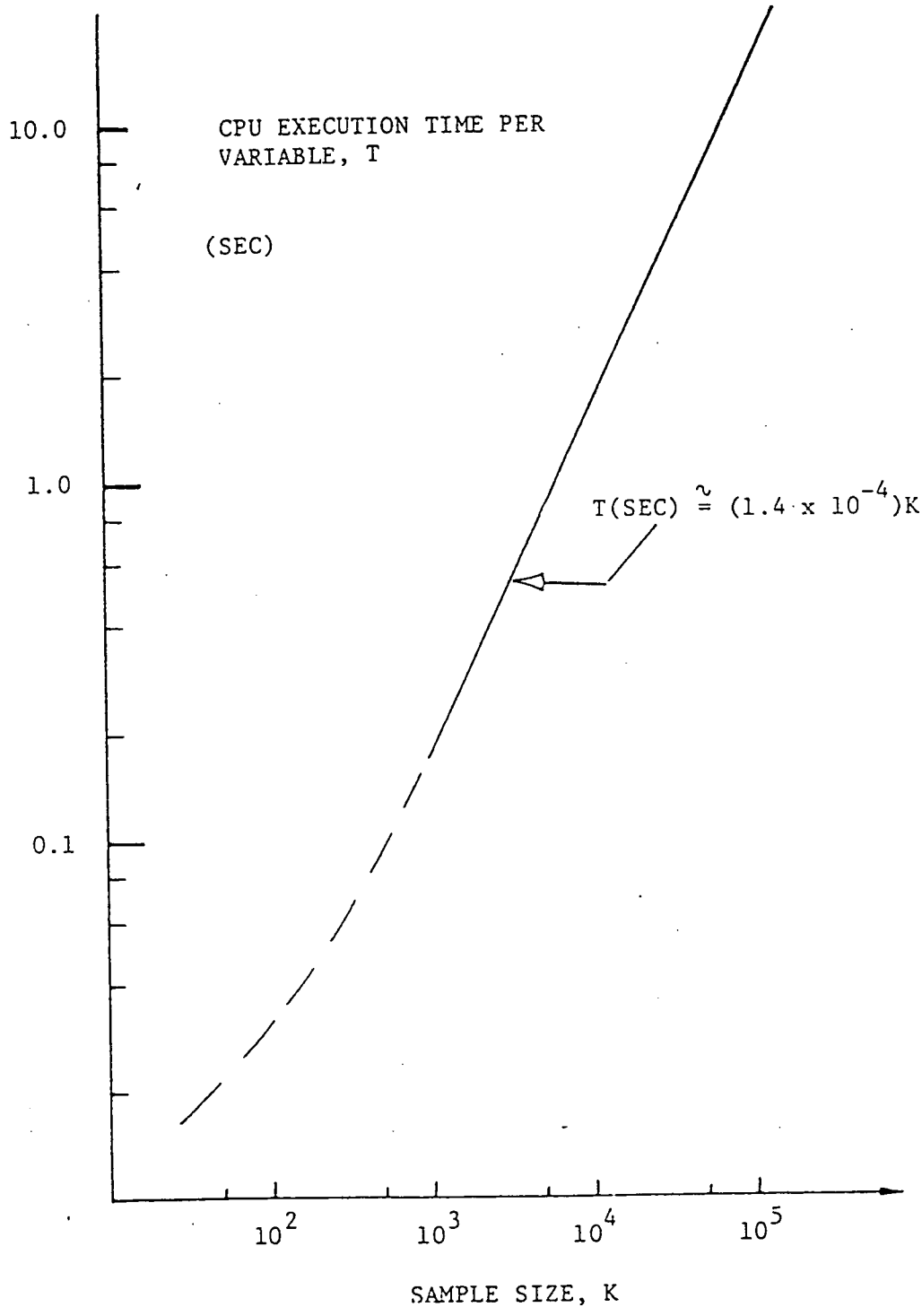


Fig. 2.5 CPU execution time per variate on CYBER 175 as a function of sample size K.

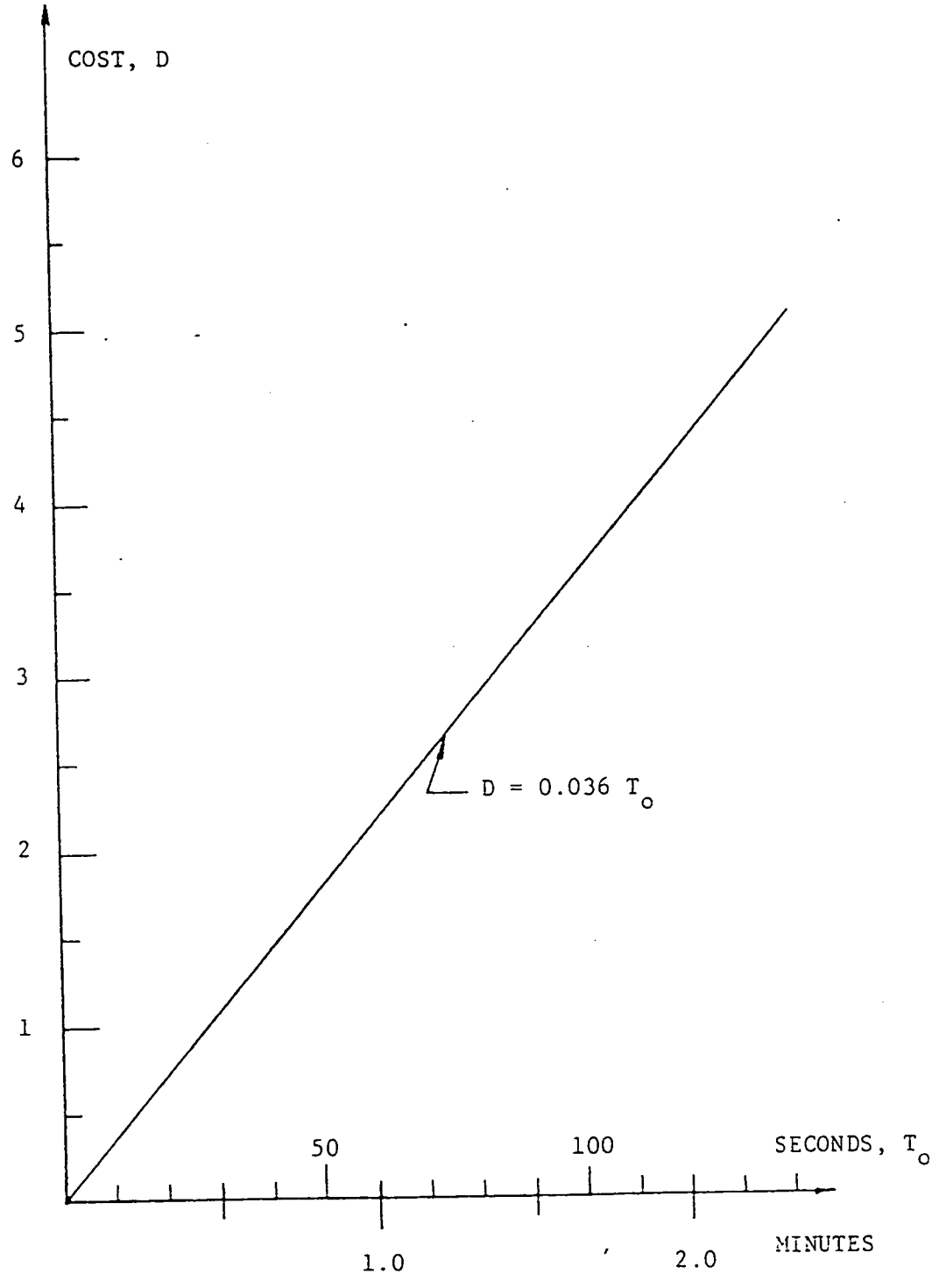


Fig. 2.6 Cost in dollars ($), D, as a function of time for the UA CYBER 175; lowest priority.

2.4 Comparison of Monte Carlo to Wu/FPI

Computational efficiency was the motivation for the development of the Wu/FPI program. It is generally known that Monte Carlo is inefficient relative to a fast probability integration method. Because the cost of conventional Monte Carlo depends upon the accuracy and probability level required, a general direct comparison can't be made. However, an example presented in the following clearly demonstrates the high cost of Monte Carlo.

Suppose that it is required to provide a Monte Carlo solution such that the 95% CI for p is within $\pm$ 10% of $\hat{p}$. The CPU execution time for the CYBER 175 can be computed from Figs. 2.3 and 2.5 for a given probability level, $\hat{p}$. This CPU time is shown in Fig. 2.7 as a function of the number of variables in $g(\underset{\sim}{X})$ for $\beta = 2$ and 3 ($p = \Phi(-\beta)$). At these levels Monte Carlo is two to three orders of magnitude more expensive than FPI. And the FPI solution is likely to be more accurate. Moreover, for smaller tail probabilities FPI gets no more expensive while Monte Carlo will break the bank.

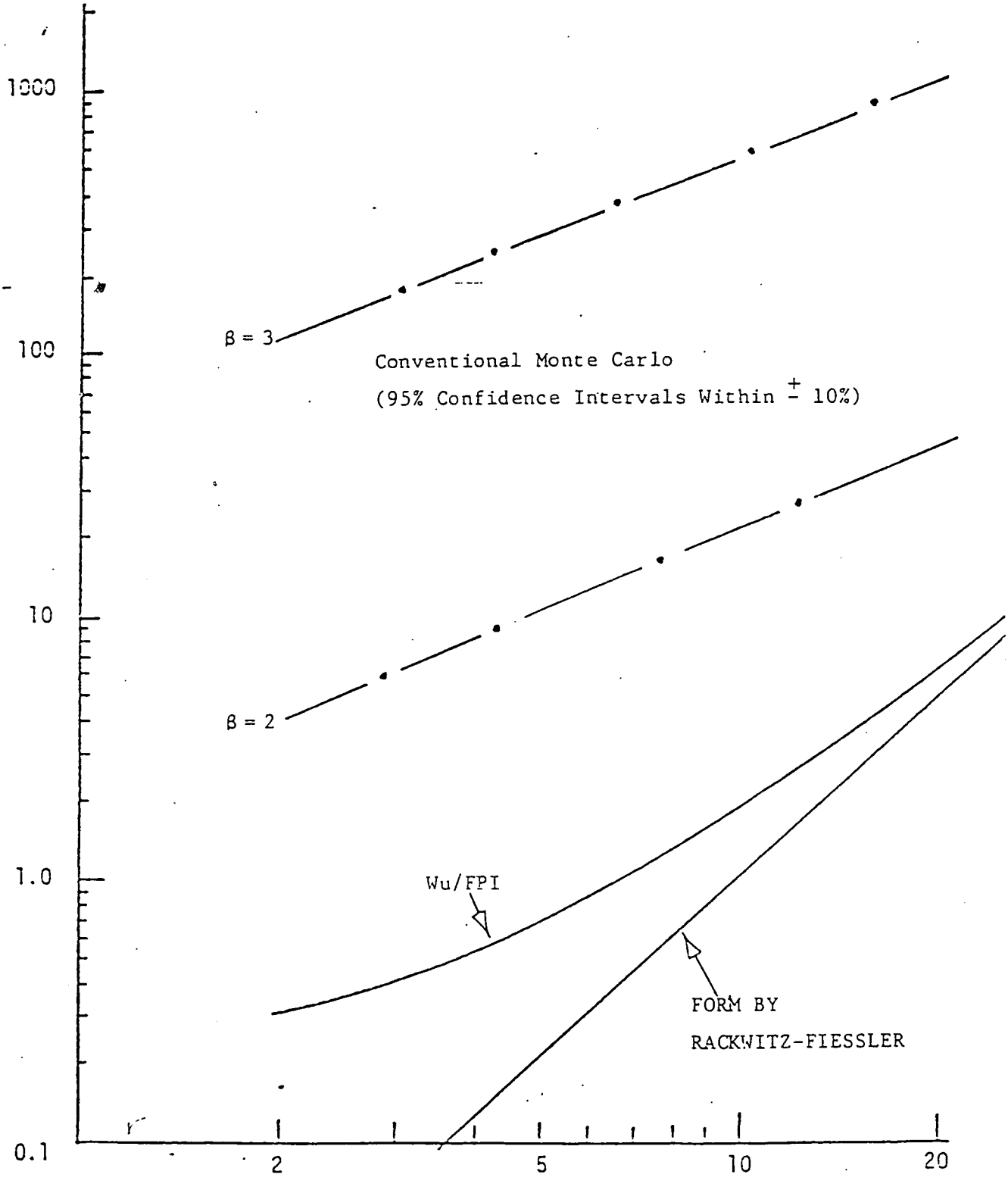
2.5 Estimating the CDF of a Random Function

2.5.1 The Empirical CDF

Conventional Monte Carlo provides capability for estimating the complete distribution function of a function of random variables. Define the random variable Z, as a function of the random vector $\underset{\sim}{X}$.

$$Z = Z(\underset{\sim}{X}) \tag{2.14}$$

Fig. 2.7 A Comparison of CPU Execution Time on the CYBER 175 Between Conventional Monte Carlo and Fast Probability Integration



A random sample of X_i; i = 1, K is used to generate a random sample of Z_i; i = 1, K. In turn, an empirical distribution function of Z can be constructed using methods of probability plotting. The empirical CDF, denoted as F_i, will be an estimate of the CDF of Z, $F_Z(z)$.

Various forms of F_i have been proposed [3, 4, 6]. The values of F_i below correspond to $Z_{(i)}$ where $Z_{(i)}$ is the ith smallest value of the random vector $\underset{\sim}{Z}$. Thus, $F_i \equiv F_i(Z_{(i)})$.

1. Hazen; $F_i = \dfrac{i - 1/2}{K}$

2. Gumbel; $F_i = \dfrac{i}{K + 1}$

3. Median ranks, $F_i = \dfrac{i - 0.3}{n + 0.4}$

Through prior experience on extensive Monte Carlo simulation, this author has found that the Hazen formula consistently provides "good estimates" of F_Z.

2.5.2 The Sort Routine

To construct the empirical CDF it is required to sort the random sample $\underset{\sim}{Z}$ to obtain an ordered sample $\underset{\sim}{Z}_o$. Let $Z_{(i)}$ denote the ith smallest value.

The routine used in this Monte Carlo code is program QUICKSORT which is considered to be the fastest available [7]. A description of QUICKSORT is given in Appendix C . The Fortran statements for this code are provided in the program listing in Appendix B .

CPU time requirements for the sort routine can be relatively large for large samples. Fig. 2.8 shown CPU execution times as a function of the size of the $\underset{\sim}{Z}$ vector.

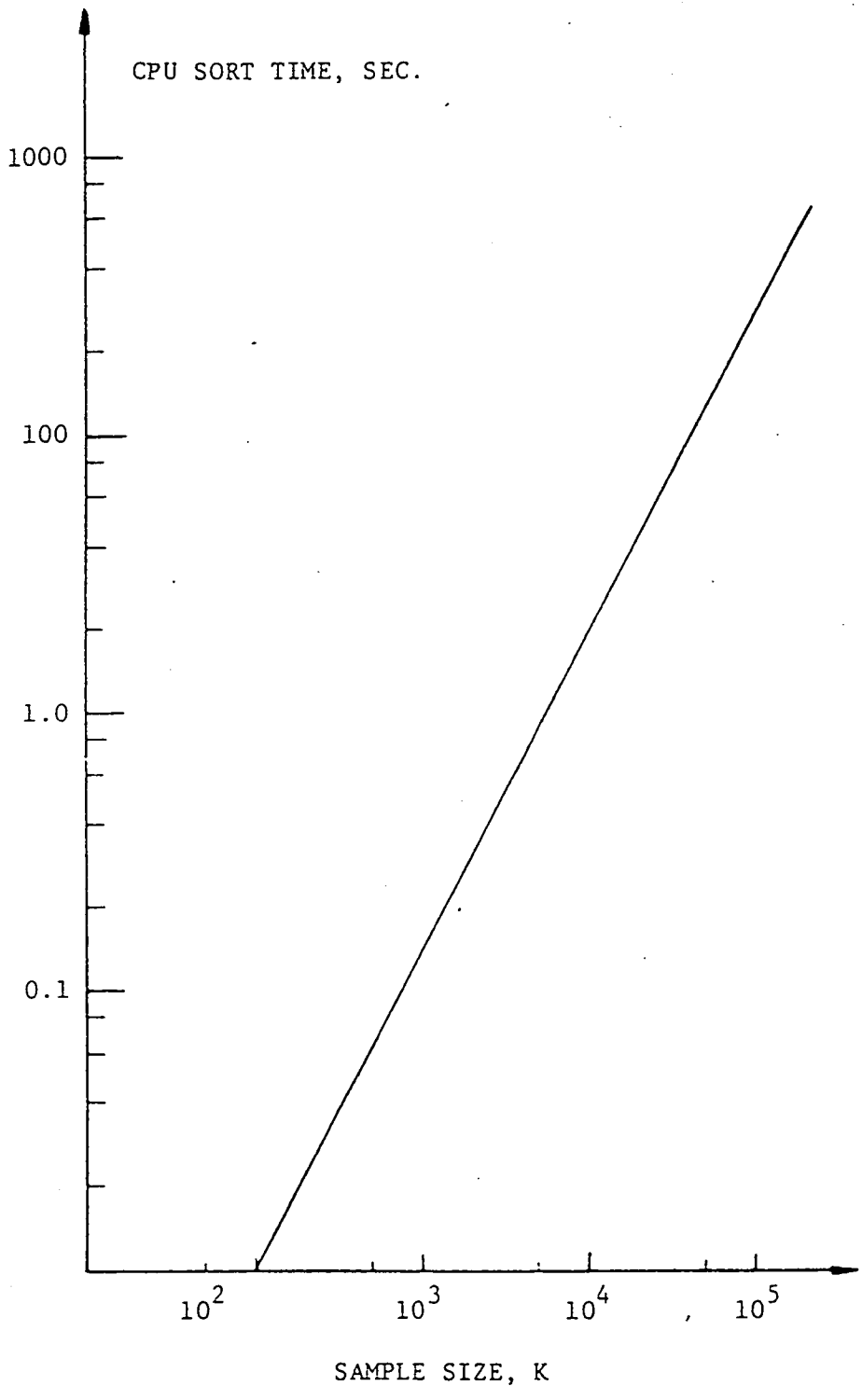


Fig. 2.8 CPU sort time (execution) as a function of sample size for the CYBER 175.

2.5.3 An Example.

Shown in Fig. 2.9 is a table of the sorted vector $Z_{(i)}$ and the corresponding F_i for the example of Fig. 2.1. This is the data required for plotting. The empirical CDF of Fig. 2.10 was done by hand, but in general such graphs can be automated using a computer graphics package.

SORTED VALUES OF Z AND THE EMPIRICAL CDF

= 1	.32159E+01	.40876E+01	.42831E+01	.44764E+01	.45626E+01
= 6	.48457E+01	.48984E+01	.50586E+01	.56150E+01	.59102E+01
= 11	.59944E+01	.60426E+01	.66202E+01	.68500E+01	.69210E+01
= 16	.69827E+01	.70597E+01	.70685E+01	.70780E+01	.71004E+01
= 21	.76156E+01	.79653E+01	.83861E+01	.84534E+01	.84720E+01
= 26	.87304E+01	.87709E+01	.87964E+01	.88850E+01	.89137E+01
= 31	.90619E+01	.90971E+01	.91454E+01	.92372E+01	.92557E+01
= 36	.92816E+01	.92823E+01	.93259E+01	.95770E+01	.95829E+01
= 41	.95862E+01	.95993E+01	.96380E+01	.98157E+01	.98782E+01
= 46	.10054E+02	.10115E+02	.10137E+02	.10256E+02	.10370E+02
= 51	.10376E+02	.10581E+02	.10607E+02	.10631E+02	.10644E+02
= 56	.10712E+02	.10771E+02	.10773E+02	.10791E+02	.10846E+02
= 61	.10856E+02	.10874E+02	.10958E+02	.11125E+02	.11162E+02
= 66	.11191E+02	.11246E+02	.11344E+02	.11409E+02	.11616E+02
= 71	.11730E+02	.11760E+02	.11802E+02	.11912E+02	.11933E+02
= 76	.12122E+02	.12140E+02	.12284E+02	.12413E+02	.12573E+02
= 81	.12667E+02	.12803E+02	.12844E+02	.12867E+02	.12873E+02
= 86	.12893E+02	.12963E+02	.13042E+02	.13131E+02	.13142E+02
= 91	.13273E+02	.13297E+02	.13361E+02	.13638E+02	.13709E+02
= 96	.13943E+02	.14797E+02	.14983E+02	.15123E+02	.15305E+02

: 1	.50000E-02	.15000E-01	.25000E-01	.35000E-01	.45000E-01
: 6	.55000E-01	.65000E-01	.75000E-01	.85000E-01	.95000E-01
: 11	.10500E+00	.11500E+00	.12500E+00	.13500E+00	.14500E+00
: 16	.15500E+00	.16500E+00	.17500E+00	.18500E+00	.19500E+00
: 21	.20500E+00	.21500E+00	.22500E+00	.23500E+00	.24500E+00
: 26	.25500E+00	.26500E+00	.27500E+00	.28500E+00	.29500E+00
31	.30500E+00	.31500E+00	.32500E+00	.33500E+00	.34500E+00
36	.35500E+00	.36500E+00	.37500E+00	.38500E+00	.39500E+00
41	.40500E+00	.41500E+00	.42500E+00	.43500E+00	.44500E+00
46	.45500E+00	.46500E+00	.47500E+00	.48500E+00	.49500E+00
51	.50500E+00	.51500E+00	.52500E+00	.53500E+00	.54500E+00
56	.55500E+00	.56500E+00	.57500E+00	.58500E+00	.59500E+00
61	.60500E+00	.61500E+00	.62500E+00	.63500E+00	.64500E+00
66	.65500E+00	.66500E+00	.67500E+00	.68500E+00	.69500E+00
71	.70500E+00	.71500E+00	.72500E+00	.73500E+00	.74500E+00
76	.75500E+00	.76500E+00	.77500E+00	.78500E+00	.79500E+00
81	.80500E+00	.81500E+00	.82500E+00	.83500E+00	.84500E+00
86	.85500E+00	.86500E+00	.87500E+00	.88500E+00	.89500E+00
91	.90500E+00	.91500E+00	.92500E+00	.93500E+00	.94500E+00
96	.95500E+00	.96500E+00	.97500E+00	.98500E+00	.99500E+00

2.9 Sorted Z_i and corresponding empirical CDF for the example of Fig. 2.1

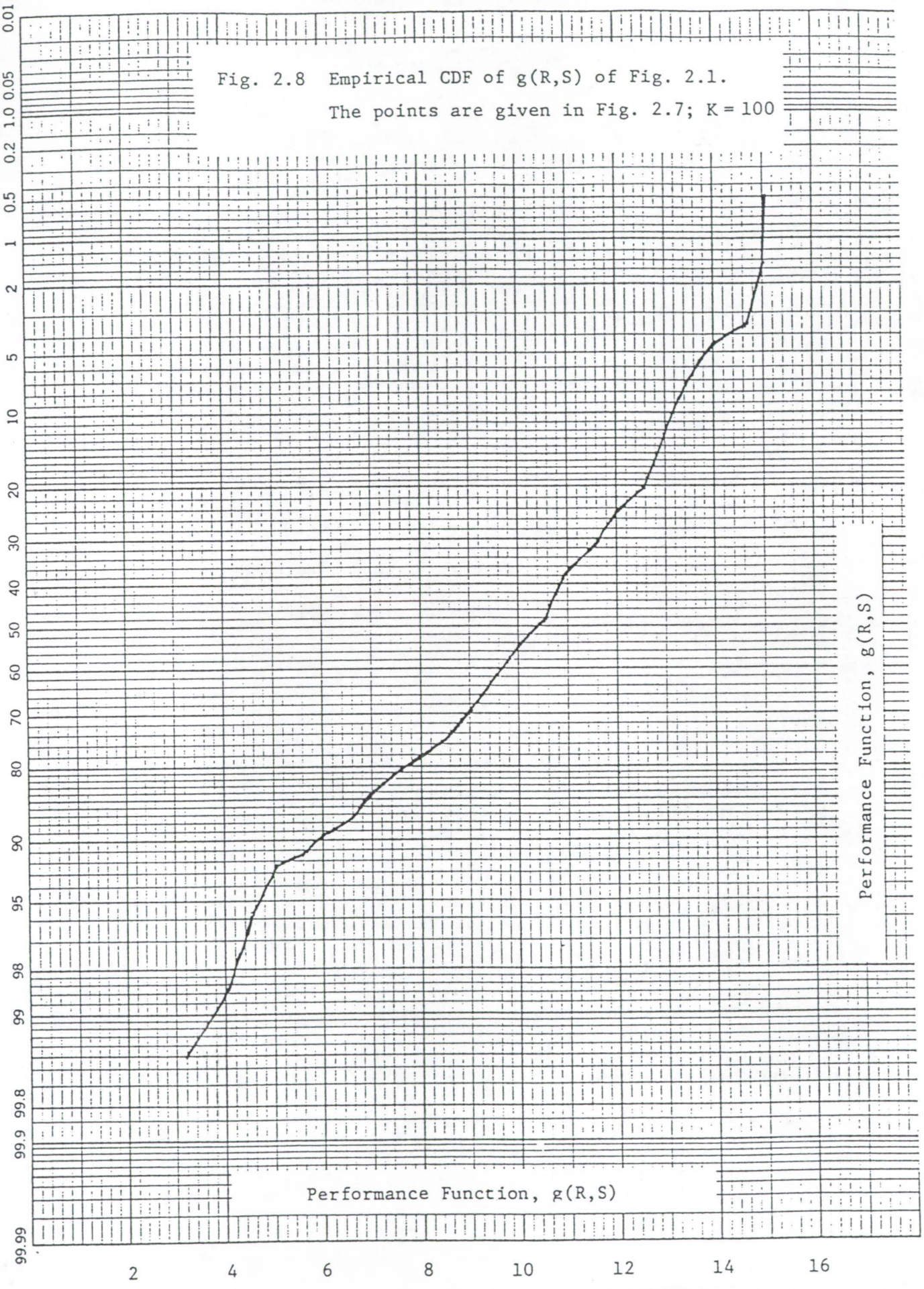


Fig. 2.8 Empirical CDF of g(R,S) of Fig. 2.1. The points are given in Fig. 2.7; K = 100

3.0 THE VARIANCE REDUCTION METHOD

3.1 Preliminary Remarks

The variance of Monte Carlo estimators can be reduced, relative to straightforward sampling of Chapt. 2.0, by appropriate operations with negatively correlated samples. Ang and Tang [1] present several examples which demonstrate dramatic improvements in efficiency realized by variance reduction methods.

A variance reduction computer program, tailored for structural mechanics analysis by providing point probability estimates of functions of random variables has been developed. The listing is given in Appendix D. To assess performance, the program has been exercised on several examples. Results presented in Section 3.6 show dramatic improvement of variance reduction over conventional Monte Carlo in some cases. In other cases, the improvement is only modest. Some general conclusions are presented in Section 3.7. For the most part however, for a given problem it is difficult to predict how much improvement one can expect with variance reduction.

3.2 The Essence of Variance Reduction

The goal of analysis is to estimate

$$p = P[h(\underset{\sim}{X}) < h_o] \tag{3.1}$$

Suppose $\hat{p}$ and $\hat{p}'$ are two unbiased estimates of p. (The method for obtaining a point estimate of p is described in Sec. 3.4 below.) The two estimators may be combined to form another estimator

$$\bar{p} = \frac{1}{2}(\hat{p} + \hat{p}') \tag{3.2}$$

The expected value of $\bar{p}$ is,

$$E(\bar{p}) = \frac{1}{2}[E(\hat{p}) + E(\hat{p}')] = p \tag{3.3}$$

which means that $\bar{p}$ is an unbiased estimator.

The corresponding variance is

$$V(\bar{p}) = \frac{1}{4}[V(\hat{p}) + V(\hat{p}') + 2\ \mathrm{Cov}\ (\hat{p}, \hat{p}')] \tag{3.4}$$

If p and p' are statistically independent, for example, based on two separate and independent sets of random numbers,

$$V(\bar{p}) = \frac{1}{4}\ [V(\hat{p}) + V(\hat{p}')] \tag{3.6}$$

Thus, the accuracy of the estimator $\bar{p}$ can be improved over that of the independent case if $\hat{p}$ and $\hat{p}'$ are <u>negatively</u> correlated. Ang and Tang cite several examples (no structural analysis) where variance reduction can provide a dramatic improvement in efficiency of probability estimation [1].

An estimate of p is obtained by several samples, $\bar{p}_i$; i = 1,K.

$$p_E = \frac{1}{K} \sum_{i=1}^{K} \bar{p}_i \tag{3.7}$$

all $\bar{p}_i$ are independent. Note that p_E will approach normality as $K \to \infty$ as a consequence of the central limit theorem.

The mean and variance of p_E are,

$$E(p_E) = p \tag{3.8}$$

$$V(p_E) = \sigma_p^2/K \tag{3.9}$$

where σ_p^2 is estimated as,

$$s_p^2 = \frac{1}{K-1} \sum_{i=1}^{K} (\bar{p}_i - p_E)^2 \tag{3.10}$$

3.3 How to Obtain Negatively Correlated Samples

Suppose that the uniformly distributed variate u_i is used to generate a number x_i from a given distribution (See Appendix A). Then the uniform variate $u_i' = 1 - u_i$ will produce an x_i' such that x_i and x_i' will be negatively correlated. The u_i' are called "antithetic" variates.

And in general, if $u_1, u_2, \ldots u_n$ is used to generate $\hat{p}$, and $1 - u_1$, $1 - u_2, \ldots 1 - u_n$ is used to generate $\hat{p}'$, then $\hat{p}$ and $\hat{p}'$ will be negatively correlated.

Such a procedure works well when the integral transform is used, e.g., Weibull, EVD. One uniform variate u_i is used to generate one x_i. But where Box-Muller is used to generate normal variates, two u_i are chosen (See Appendix A). While the resulting x_i and x_i' will be negatively correlated, the correlation coefficient will not be -1.0. An improvement can be made by choosing x_i' as a "mirror image" of x_i in the distributions. This can be done by

$$x_i' = 2\mu - x_i \tag{3.11}$$

where μ is the mean of X.

3.4 How to Obtain Point Probability Estimates

3.4.1 The Two Variable Case

The structural reliability problem in which p is the probability of failure will be used to illustrate how $\hat{p}$ and $\hat{p}'$ are obtained. Consider the design case where the two variables are R (strength) and S (stress). Estimate p, where

$$p = P[R - S \leq 0] \tag{3.12}$$

Both R and S are random variables whose density functions are shown in Fig. 3.1. First S, having been identified as the variable having the largest variance, is the "reference." A random variate R_i is sampled from the other factor, R. An estimate of p is

$$\hat{p}_i = P(S > R_i)$$

$$= 1 - F_S(R_i) \tag{3.13}$$

where F_S is the CDF of S.

It should now be apparent why sampling is done on the smallest variance term. $\hat{p}$ is a "good" estimate of p if the distribution is narrow, and is exact as $\sigma_R \to 0$.

Now the antithetic variate R_i' is sampled as described above. Because it is negatively correlated to R_i, its position relative to R_i will be as shown in Fig. 3.2. Then,

$$\hat{p}_i' = P(S > R_i')$$

$$= 1 - F_S(R_i') \tag{3.14}$$

and the ith estimate of p is

$$\bar{p}_i = \frac{1}{2}(\hat{p}_i + \hat{p}_i') \tag{3.15}$$

As a second example, consider again the case where R and S are the basic variables, but now where $\sigma_R < \sigma_S$. In this case, R would be the reference variable. Random points S_i and the antithetic variate S_i' are sampled from S. The estimates now are,

$$\hat{p}_i = F_R(S_i) \tag{3.16}$$

$$\hat{p}_i' = F_R(S_i')$$

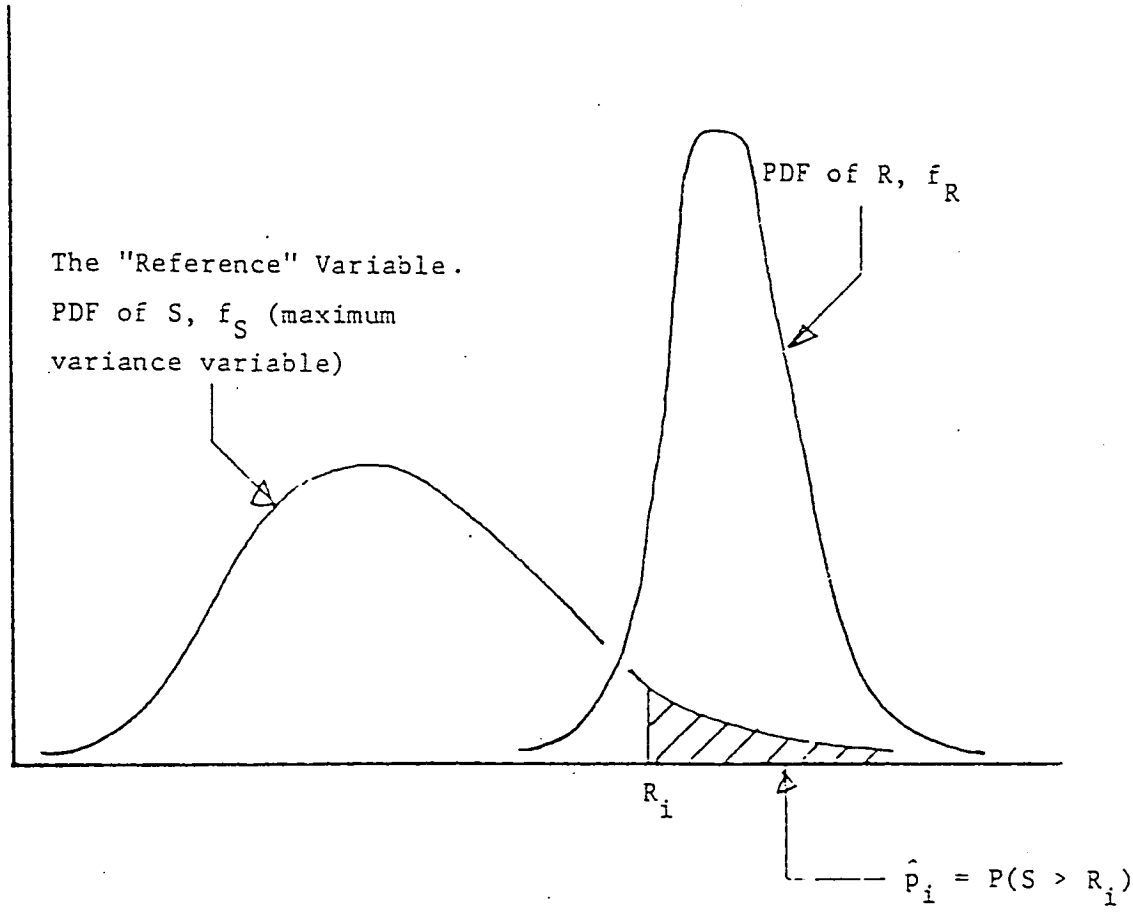


Fig. 3.1 Estimate of p using one point sampled from the minimum variance variable.

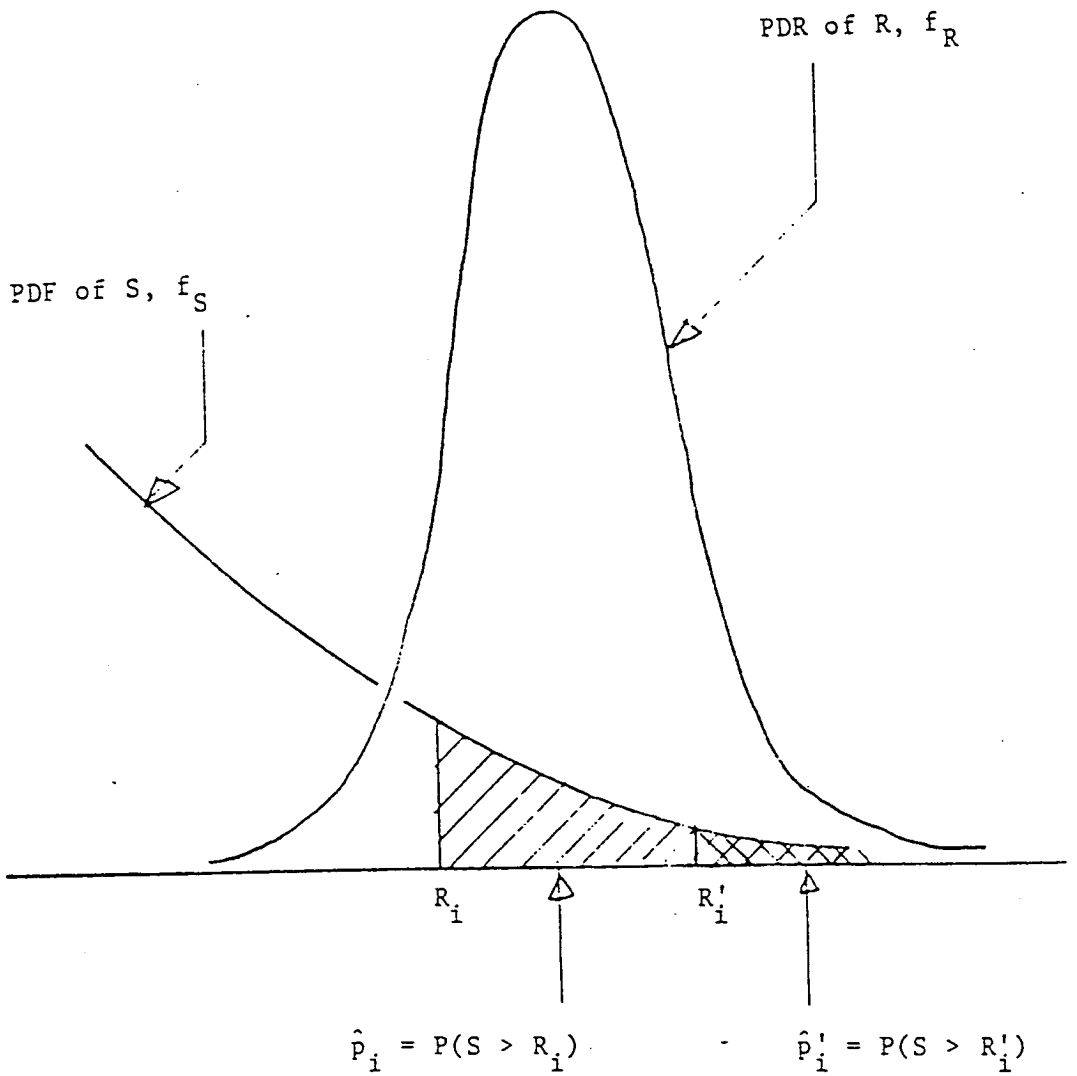


Fig. 3.2 Estimates of p using a point R_i sampled from R and the antithetic variate of R_i, denoted as R_i'

Thus, it is seen that the variable type (stress or strength) must be identified to obtain the proper form for computing estimates.

Fig. 3.2 shows why negatively correlated variables tend to provide good estimates. Being on both sides of a distribution, R_i and R_i' combine to produce an "average" estimate of p.

3.4.2 The General Case

In general, the performance function, $g(\underset{\sim}{X}) = h(\underset{\sim}{X}) - h_o$ is a non-linear function of several variables. The method of obtaining a point estimate of p is an extension of the scheme for two variables.

The reference variable is defined, not as the one having the maximum variance, but rather the one having the maximum impact. For example, if

$$g = 5R - S \tag{3.17}$$

and $\sigma_R = \sigma_S/2$, clearly the random variable, $R_1 = 5R$ will have a larger variance than S. Thus, we say that R is the maximum impact variable.

In general, the maximum impact variable can be found by estimating $\partial g/\partial X_i$ for each X_i. The maximum impact variable, denoted as X_M, is that X_i for which $|\partial g/\partial X_i|$ is the largest.

The sign of $\partial g/\partial X_i$ identifies variable type; stress if (+) and strength if (-). As indicated above, the "type" of X_M must be known to choose the appropriate form for estimating p (e.g., Eqs. 3.13 and 3.16).

The estimates $\hat{p}$ and $\hat{p}'$ proceed as follows. Sample all variables but X_M. Let $g(\underset{\sim}{X}) = 0$, and solve for x_M (this is done by the secant method in the program).

$$x_M = h(\underset{\sim}{x}_o) \tag{3.18}$$

where $\underset{\sim}{x}_o$ is the vector of sampled $\underset{\sim}{X}$ minus X_M.

The estimate of p is,

$$\hat{p} = \begin{cases} F_{X_M}(x_M) & \text{if } X_M \text{ is a strength variable} \\ 1 - F_{X_M}(x_M) & \text{if } X_M \text{ is a stress variable} \end{cases} \tag{3.19}$$

To obtain $\hat{p}'$, the antithetic vector $\underset{\sim}{x}_o'$ of $\underset{\sim}{x}_o$ is used in Eq. 3.19.

3.5 Confidence Intervals on p

Noting that p_E is normally distributed, approximate $1 - \alpha$ confidence intervals on p can be constructed as [5],

$$p_E - \frac{z_{\alpha/2}\, s_p}{\sqrt{K}} < p < p_E + \frac{z_{\alpha/2}\, s_p}{\sqrt{K}} \tag{3.20}$$

or,

$$p_E(1 - \gamma) < p < p_E(1 + \gamma) \tag{3.21}$$

where,

$z_{\alpha/2}$ = standard normal variate (absolute value) at probability level $\alpha/2$.

$$\gamma = \frac{z_{\alpha/2}\, C_p}{\sqrt{K}} \tag{3.22}$$

$$C_p = s_p / p_E \tag{3.23}$$

The UA variance reduction program chooses K to produce a specific confidence interval. For example, if you want to sample until the 95% confidence intervals are $\pm$ 10% of p_E,

$$\gamma = 0.10 \qquad z_{\alpha/2} = 1.64 \tag{3.24}$$

and solving Eq. 3.22 for K,

$$K > \frac{z_{\alpha/2}^2 \, C_p^2}{\gamma} = 269 \, C_p^2 \qquad (3.25)$$

To find C_p, an initial sample of K = 1000 is chosen and an estimate of C_p is obtained. Then if K < 1000 in Eq. 3.25, the process is terminated with narrower confidence intervals than requested. If K > 1000, the program will continue to sample to that value.

3.6 The Variance Reduction Monte Carlo Program

A flow diagram which outlines the logic of the variance reduction program is provided in Fig. 3.3. Sample output of the program is shown in Fig. 3.4 with some commentary.

Two versions of the program have been developed. An interactive version (IVARED) runs on the IBM PC/XT. Program VARED runs on the VAX or CYBER 175. A listing of VARED is given in Appendix D.

3.7 Examples of the Performance of VARED

Twelve examples of the use of VARED to produce point probability estimate are provided in Tables 3.1 through 3.12. Point estimates by VARED are compared to the exact solution (closed form or POFAIL) if available. The exact solution, provided by program POFAIL, is employed for performance functions involving two variables. For larger problems, Wu/FPI is used. For the VARED solutions, 95% confidence intervals (α = 5%) are specified along with γ = 0.10.

To compare variance reduction with conventional Monte Carlo, sample size requirements and CPU time for the latter are extracted from Figs. 2.4 and 2.5 and are presented in the tables.

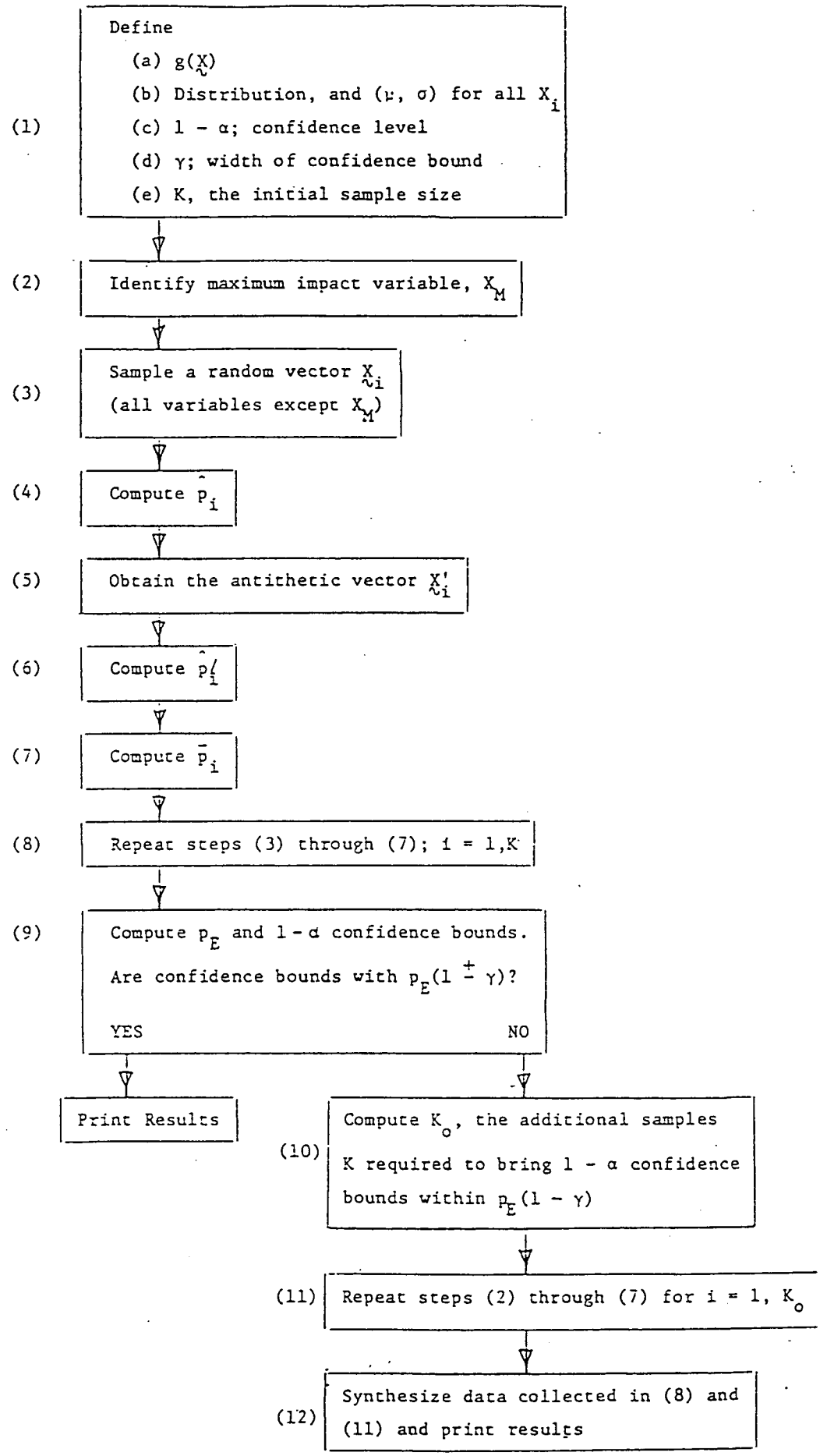


Fig. 3.3 An outline of the variance reduction Monte Carlo program

Fig. 3.4 An example of the output of the variance reduction Monte Carlo Program with commentary

MONTE CARLO SOLUTION

LIMIT STATE FUNCTION : G=R-DSQRT(300.*P**2+1.92*T**2)

SAMPLE SIZE = 1000

NUMBER OF RANDOM VARIABLES = 3

This value is arbitrary; it is the size of the first sample used to estimate the total required sample size, K

CONFIDENCE INTERVAL = 95.00 %

GAMMA = .10

Ensures that 95% confidence intervals on p will be within $\pm$ 10% of the estimator, p_E

MAX. IMPACT VARIABLE = X(1)

VARIABLE TYPE IS STRENGTH

RANDOM VARIABLES

VARIABLE	DISTRIBUTION	MEAN	STD DEV
R	WEIBULL	.48000E+02	.30000E+01
P	LOG	.98700E+00	.16000E+00
T	EVD	.20000E+02	.20000E+01

ESTIMATE OF P = .16043E-02

This is the first estimate of p

95.00 % CONFIDENCE INTERVALS ARE

PL = .11725E-02 PU = .20360E-02

Note that 95% confidence intervals exceed $\pm$ 10%. Thus, a larger K is required. (See below)

STATISTICS OF P :

MEAN = .16043E-02

(Continued on next page)

```
STD DEV =    .69662E-02
MEDIAN  =    .36004E-03
COV     =    .43422E+01

K FOR GAMMA =    .10 IS    7244

ESTIMATE OF P =    .18030E-02
95.00 % CONFIDENCE INTERVALS ARE
PL =    .15509E-02      PU =    .20550E-02

STATISTICS OF P :
MEAN    =    .18348E-02
STD DEV =    .11456E-01
MEDIAN  =    .29017E-03
COV     =    .62436E+01

YOU HAVE ANOTHER DATA SET ?(Y/N)
```

Based on the first sample of K = 1(this is the total K required for th desired confidence intervals. K is computed from Eq. 3.25 which requir C_p. This is why the first sample c 1000 is taken.

Note that the confidence intervals do not qui meet the specifications. This is because the original estimate of $C_p = 4.34$ was small rela to the improved estimate of $C_p = 6.24$

Note: The size of the sample required K depends upon C_p (Eq. 3.25). In this problem C_p is relatively large implying that a relatively large K is required. This same problem is presented in Table 3.7.

Table 3.1 Example of the Performance of a Variance Reduction Monte Carlo Program; EXAMPLE 1

DEMONSTRATING THE PERFORMANCE OF THE UA VARIANCE REDUCTION MONTE CARLO PROGRAM

EXAMPLE 1

PERFORMANCE FUNCTION: $g = R - S$

Variable	Type	Mean/Median*	Std. Dev./ COV*
R	N	50	5
S	N	20	12

RESULTS:

	Probability of Failure	Total CPU-Time(b)	Sample Size, K(c)
Exact(a) Wu/FPI	1.051E-2		
Monte Carlo Variance Reduction(d)	1.118E-2	2.04	160
Monte Carlo Conventional (Bernoulli parameter)(e)		11.2	5E4

Notes:

(a) Exact value using POFAIL if two variables. If more than two, Wu/FPI is used; the exact should be within 5% of this value.

(b) CYBER 175

(c) The number of $\bar{p}_i$ for variance reduction and the number of Z_i for conventional. The values are not directly comparable.

(d) 95% confidence intervals within $\pm$ 10% of p_E

(e) Same confidence interval as variance reduction.

Table 3.2 Example of the Performance of a Variance Reduction Monte Carlo Program; EXAMPLE 2

DEMONSTRATING THE PERFORMANCE OF THE UA VARIANCE REDUCTION MONTE CARLO PROGRA

EXAMPLE 2

PERFORMANCE FUNCTION: $g = R - S$

Variable	Type	Mean/Median*	Std. Dev./ COV*
R	LN	50 *	0.2 *
S	LN	20 *	0.2 *

RESULTS:

	Probability of Failure	Total CPU-Time(b)	Sample Size, K(c)
Exact(a) Wu/FPI	5.347E-4		
Monte Carlo Variance Reduction(d)	5.072E-4	13.78	11589
Monte Carlo Conventional (Bernoulli parameter)(e)		238.9	1.122E6

Notes:

(a) Exact value using POFAIL if two variables. If more than two, Wu/FPI is used; the exact should be within 5% of this value.

(b) CYBER 175

(c) The number of $\bar{p}_i$ for variance reduction and the number of Z_i for conventional. The values are not directly comparable.

(d) 95% confidence intervals within $\pm$ 10% of p_E

(e) Same confidence interval as variance reduction.

Table 3.3 Example of the Performance of a Variance Reduction Monte Carlo Program; EXAMPLE 3

DEMONSTRATING THE PERFORMANCE OF THE UA VARIANCE REDUCTION MONTE CARLO PROGRAM

EXAMPLE 3

PERFORMANCE FUNCTION: $g = R - S$

Variable	Type	Mean/Median*	Std. Dev./ COV*
R	WEI	4.5	0.45
S	FRE	3.0	0.30

RESULTS:

	Probability of Failure	Total CPU-Time(b)	Sample Size, K(c)
Exact(a) Wu/FPI	1.0933E-2		
Monte Carlo Variance Reduction(d)	1.0914E-2	4.066	2535
Monte Carlo Conventional (Bernoulli parameter)(e)		9.634	35481

Notes:

(a) Exact value using POFAIL if two variables. If more than two, Wu/FPI is used; the exact should be within 5% of this value.

(b) CYBER 175

(c) The number of $\bar{p}_i$ for variance reduction and the number of Z_i for conventional. The values are not directly comparable.

(d) 95% confidence intervals within $\pm$ 10% of p_E

(e) Same confidence interval as variance reduction.

Table 3.4 Example of the Performance of a Variance Reduction Monte Carlo Program; EXAMPLE 4

DEMONSTRATING THE PERFORMANCE OF THE UA VARIANCE REDUCTION MONTE CARLO PROGRA

EXAMPLE 4

PERFORMANCE FUNCTION: $g = R - S^2$

Variable	Type	Mean/Median*	Std. Dev./ COV*
R	WEI	20	4.0
S	FRE	3	0.6

RESULTS:

	Probability of Failure	Total CPU-Time(b)	Sample Size, K(c)
Exact(a) Wu/FPI	4.272E-2		
Monte Carlo Variance Reduction(d)	4.0511E-2	3.568	1864
Monte Carlo Conventional (Bernoulli parameter)(e)		3.689	9441

Notes:

(a) Exact value using POFAIL if two variables. If more than two, Wu/FPI is used; the exact should be within 5% of this value.

(b) CYBER 175

(c) The number of $\bar{p}_i$ for variance reduction and the number of Z_i for conventional. The values are not directly comparable.

(d) 95% confidence intervals within $\pm$ 10% of p_E

(e) Same confidence interval as variance reduction.

Table 3.5 Example of the Performance of a Variance Reduction Monte Carlo Program; EXAMPLE 5

DEMONSTRATING THE PERFORMANCE OF THE UA VARIANCE REDUCTION MONTE CARLO PROGRAM

EXAMPLE 5

PERFORMANCE FUNCTION: $g = R - S$

Variable	Type	Mean/Median*	Std. Dev./ COV*
R	WEI	20	2.0
S	EVD	10	2.0

RESULTS:

	Probability of Failure	Total CPU-Time(b)	Sample Size, K(c)
Exact(a) Wu/FPI	2.8573E-3		
Monte Carlo Variance Reduction(d)	2.6179E-3	10.881	11362
Monte Carlo Conventional (Bernoulli parameter)(e)		36.157	152230

Notes:

(a) Exact value using POFAIL if two variables. If more than two, Wu/FPI is used; the exact should be within 5% of this value.

(b) CYBER 175

(c) The number of $\bar{p}_i$ for variance reduction and the number of Z_i for conventional. The values are not directly comparable.

(d) 95% confidence intervals within $\pm$ 10% of p_E

(e) Same confidence interval as variance reduction.

Table 3.6 Example of the Performance of a Variance Reduction Monte Carlo Program; EXAMPLE 6

DEMONSTRATING THE PERFORMANCE OF THE UA VARIANCE REDUCTION MONTE CARLO PROGRA

EXAMPLE 6

PERFORMANCE FUNCTION: $g = \frac{\Delta A}{B^m 0} - T_S + \frac{\Delta A}{B^{-3} 0.2779} - 6.3E8$

Variable	Type	Mean/Median *	Std. Dev./ COV*
Δ	LN	1.0*	0.3*
A	WEI	4.3Eg	0.5*
B	LN	0.9*	0.25*

RESULTS:

	Probability of Failure	Total CPU-Time(b)	Sample Size, K(c)
Exact(a) Wu/FPI	1.901E-3		
Monte Carlo Variance Reduction(d)	1.7958E-3	3.643	1437
Monte Carlo Conventional (Bernoulli parameter)(e)		68.3616	199526

Notes:

(a) Exact value using POFAIL if two variables. If more than two, Wu/FPI is used; the exact should be within 5% of this value.

(b) CYBER 175

(c) The number of $\bar{p}_i$ for variance reduction and the number of Z_i for conventional. The values are not directly comparable.

(d) 95% confidence intervals within $\pm$ 10% of p_E

(e) Same confidence interval as variance reduction.

Table 3.7 Example of the Performance of a Variance Reduction Monte Carlo Program; EXAMPLE 7

DEMONSTRATING THE PERFORMANCE OF THE UA VARIANCE REDUCTION MONTE CARLO PROGRAM

EXAMPLE 7

PERFORMANCE FUNCTION: $g = R - \sqrt{300\ P^2 + 1.92 \cdot T^2}$

Variable	Type	Mean/Median*	Std. Dev./ COV*
R	WEI	48.0	3.0
P	LN	0.987*	0.16*
T	EVD	20.0	2.0

RESULTS:

	Probability of Failure	Total CPU-Time(b)	Sample Size, K(c)
Exact(a) Wu/FPI	0.0018		
Monte Carlo Variance Reduction(d)	0.0018208	16.375	12734
Monte Carlo Conventional (Bernoulli parameter)(e)		74.4186	211349

Notes:

(a) Exact value using POFAIL if two variables. If more than two, Wu/FPI is used; the exact should be within 5% of this value.

(b) CYBER 175

(c) The number of $\bar{p}_i$ for variance reduction and the number of Z_i for conventional. The values are not directly comparable.

(d) 95% confidence intervals within $\pm$ 10% of p_E

(e) Same confidence interval as variance reduction.

Table 3.8 Example of the Performance of a Variance Reduction Monte Carlo Program; EXAMPLE 8

DEMONSTRATING THE PERFORMANCE OF THE UA VARIANCE REDUCTION MONTE CARLO PROGRA

EXAMPLE 8

PERFORMANCE FUNCTION: $g = \Delta - 1000 \frac{f_{pp}}{G(Y \cdot \Delta\varepsilon_o)^{-1.71}} + \frac{1 - f_{pp}}{H(Y \cdot \Delta\varepsilon_o)^{-1.188}}$

Variable	Type	Mean/Median *	Std. Dev./ COV *
Δ	LN	1.0*	0.3*
f_{pp}	N	0.7	0.07
G	LN	0.222*	0.4*
T	LN	1.0*	0.15*
$\Delta\varepsilon_o$	EVD	0.0005	0.00008
H	LN	1.673*	0.4*

RESULTS:

	Probability of Failure	Total CPU-Time(b)	Sample Size, K(c)
Exact(a) Wu/FPI	1.002E-2		
Monte Carlo Variance Reduction(d)	9.8814E-3	14.822	4401
Monte Carlo Conventional (Bernoulli parameter)(e)		30.7564	39810

Notes:

(a) Exact value using POFAIL if two variables. If more than two, Wu/FPI is used; the exact should be within 5% of this value.

(b) CYBER 175

(c) The number of $\bar{p}_i$ for variance reduction and the number of Z_i for conventional. The values are not directly comparable.

(d) 95% confidence intervals within $\pm$ 10% of p_E

(e) Same confidence interval as variance reduction.

Table 3.9 Example of the Performance of a Variance Reduction Monte Carlo Program; EXAMPLE 9a

DEMONSTRATING THE PERFORMANCE OF THE UA VARIANCE REDUCTION MONTE CARLO PROGRAM

EXAMPLE 9a

PERFORMANCE FUNCTION: $g = R - S$

Variable	Type	Mean/Median*	Std. Dev./ COV*
R	LN	20.0*	0.2*
S	LN	10.0*	0.2*

RESULTS:

	Probability of Failure	Total CPU-Time(b)	Sample Size, K(c)
Exact(a) Wu/FPI	6.6642E-3		
Monte Carlo Variance Reduction(d)	6.4159E-3	4.75	2831
Monte Carlo Conventional (Bernoulli parameter)(e)		15.7724	59566

Notes:

(a) Exact value using POFAIL if two variables. If more than two, Wu/FPI is used; the exact should be within 5% of this value.

(b) CYBER 175

(c) The number of $\bar{p}_i$ for variance reduction and the number of Z_i for conventional. The values are not directly comparable.

(d) 95% confidence intervals within $\pm$ 10% of p_E

(e) Same confidence interval as variance reduction.

Table 3.10 Example of the Performance of a Variance Reduction Monte Carlo Program; EXAMPLE 9b

DEMONSTRATING THE PERFORMANCE OF THE UA VARIANCE REDUCTION MONTE CARLO PROGRAM

EXAMPLE 9b .

PERFORMANCE FUNCTION: $g = R - S$

Variable	Type	Mean/Median*	Std. Dev./ COV*
R	LN	22.5*	0.2*
S	LN	10.0*	0.2*

RESULTS:

	Probability of Failure	Total CPU-Time(b)	Sample Size, K(c)
Exact(a) Wu/FPI	1.89338E-3		
Monte Carlo Variance Reduction(d)	1.7434E-3	8.075	6068
Monte Carlo Conventional (Bernoulli parameter)(e)		51.44	218776

Notes:

(a) Exact value using POFAIL if two variables. If more than two, Wu/FPI is used; the exact should be within 5% of this value.

(b) CYBER 175

(c) The number of $\bar{p}_i$ for variance reduction and the number of Z_i for conventional. The values are not directly comparable.

(d) 95% confidence intervals within $\pm$ 10% of p_E

(e) Same confidence interval as variance reduction.

Table 3.11 Example of the Performance of a Variance Reduction Monte Carlo Program; EXAMPLE 9c

DEMONSTRATING THE PERFORMANCE OF THE UA VARIANCE REDUCTION MONTE CARLO PROGRAM

EXAMPLE 9c

PERFORMANCE FUNCTION: $g = R - S$

Variable	Type	Mean/Median*	Std. Dev./ COV*
R	LN	25.0*	0.2*
S	LN	10.0*	0.2*

RESULTS:

	Probability of Failure	Total CPU-Time(b)	Sample Size, K(c)
Exact(a) Wu/FPI	5.347E-4		
Monte Carlo Variance Reduction(d)	5.072E-4	13.681	11589
Monte Carlo Conventional (Bernoulli parameter)(e)		164.70	767361

Notes:

(a) Exact value using POFAIL if two variables. If more than two, Wu/FPI is used; the exact should be within 5% of this value.

(b) CYBER 175

(c) The number of $\bar{p}_i$ for variance reduction and the number of Z_i for conventional. The values are not directly comparable.

(d) 95% confidence intervals within $\pm$ 10% of p_E

(e) Same confidence interval as variance reduction.

Table 3.12 Example of the Performance of a Variance Reduction Monte Carlo Program; EXAMPLE 9d

DEMONSTRATING THE PERFORMANCE OF THE UA VARIANCE REDUCTION MONTE CARLO PROGRA

EXAMPLE 9d

PERFORMANCE FUNCTION: $g = R - S$

Variable	Type	Mean/Median*	Std. Dev./ COV*
R	LN	27.0*	0.2*
S	LN	10.0*	0.2*

RESULTS:

	Probability of Failure	Total CPU-Time(b)	Sample Size, K(c)
Exact(a) Wu/FPI	1.952665E-4		
Monte Carlo Variance Reduction(d)	2.0296E-4	20.27	17977
Monte Carlo Conventional (Bernoulli parameter)(e)		388.93	1840772

Notes:

(a) Exact value using POFAIL if two variables. If more than two, Wu/FPI is used; the exact should be within 5% of this value.

(b) CYBER 175

(c) The number of $\bar{p}_i$ for variance reduction and the number of Z_i for conventional. The values are not directly comparable.

(d) 95% confidence intervals within $\pm$ 10% of p_E

(e) Same confidence interval as variance reduction.

3.8 Comparison of Computer Costs of Variance Reduction and Conventional Monte Carlo

Example 9a, b, c, and d was designed to illustrate how computer costs increase as point probabilities become smaller, providing estimates at the same level of confidence. Figs. 3.5 and 3.6 show the relationship between CYBER 175 CPU execution time and the probability level for the conventional "Bernoulli" and the variance reduction estimates, respectively, for Example 9. Then Fig. 3.7 demonstrates how much more efficient is variance reduction for this problem. It should be noted that Figs. 3.5 through 3.7 relate only to Example 9 and cannot be presented as being characteristic of the relative behavior of the two methods.

3.9 Conclusions on Variance Reduction

Some general conclusions based on experiences exercising VARED are,

1. Variance reduction seems to outperform conventional Monte Carlo consistently. However, in some cases the improvement is dramatic, in some cases it is modest.

2. Related to item 1, it is difficult to predict computer costs. At a given confidence level, CPU time depends strongly upon the form of the performance function, the distribution of the variables, as well as the probability level.

3. To construct a CDF, it is necessary to obtain several point probability estimates, as it is using FPI. Thus, the variance reduction Monte Carlo method is not particularly effective when it is required to construct a distribution function of a response variable.

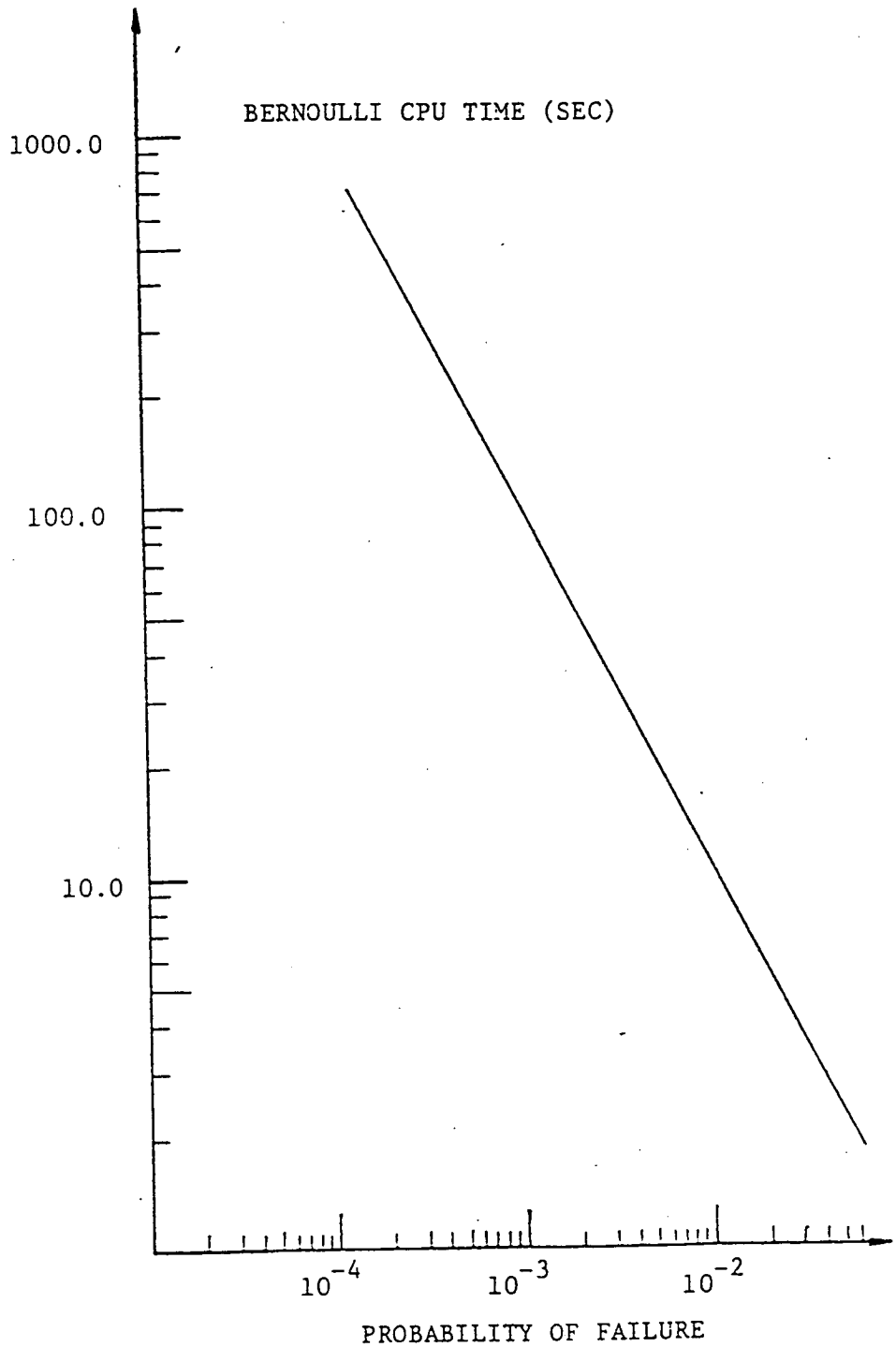


Fig. 3.5 CPU execution time for CYBER 175 for conventional Bernoulli point probability estimate; Example 9; α = 5%, γ = 10%

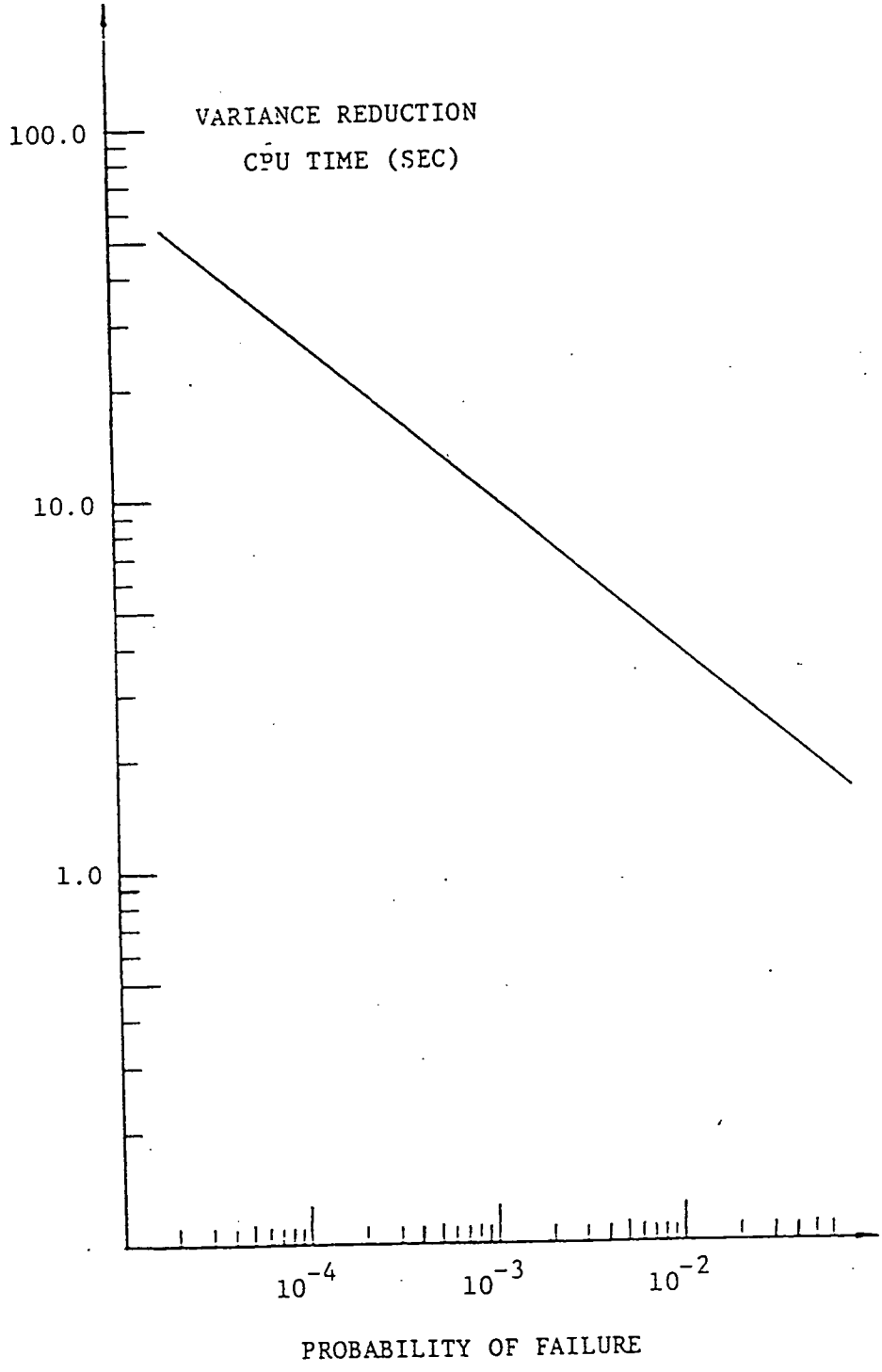


Fig. 3.6 CPU execution time for CYBER 175 for variance reduction point probability estimate; Example 9; α = 5%, γ = 10%

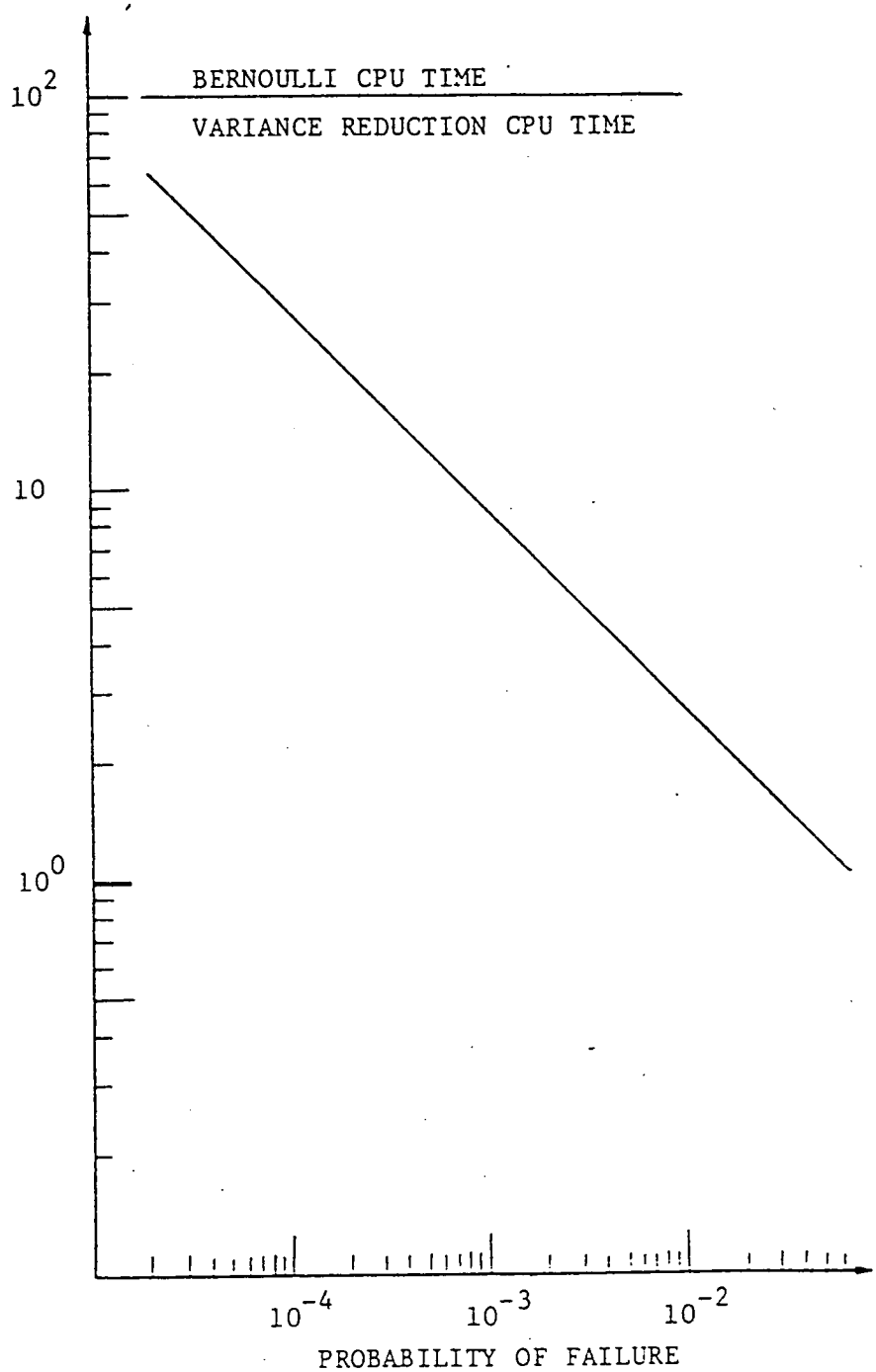


Fig. 3.7 Ratio of Bernoulli to variance reduction CPU execution time for CYBER 175 for point probability estimate; Example 9; 95% C.I.γ = 10%

4.0 DIRECT EVALUATION OF THE PROBABILITY OF FAILURE INTEGRAL

4.1 Preliminary Remarks

Consider the multidimensional integral for p_f.

$$p_f = \int_{\Omega} f_{\underset{\sim}{X}}(\underset{\sim}{x})dx \tag{4.1}$$

where $\underset{\sim}{X}$ is a vector of n random variables and Ω is region of failure in X-space.

Standard methods of numerical integration (e.g., Simpson's rule) are efficient for a one and two dimensional integral. But when n exceeds two, these methods are much more difficult to apply. Monte Carlo integration becomes more attractive for $n > 2$.

4.2 The Mean Value Method Used for a Single Random Variable

Consider the random variable X. Let

$$p_f = P[X \leq a] \tag{4.2}$$

$$p_f \equiv I = \int_0^a f(x)dx$$

The density function f(x) is shown in Fig. 4.1. But upon dividing the interval (0, a) into J equal increments, Δx_i, the integral can be approximated as

$$I = \sum_{i=1}^{J} \Delta x_i \, f(x_i) \tag{4.3}$$

This summation can be approximated by a Monte Carlo approach. Define a sampling interval (c, d). In the example of Fig. 4.1, (c, d) could be chosen as (0, a). But in general, c should be chosen so that the area below is "very small." And d should be chosen such that $d \geq a$.

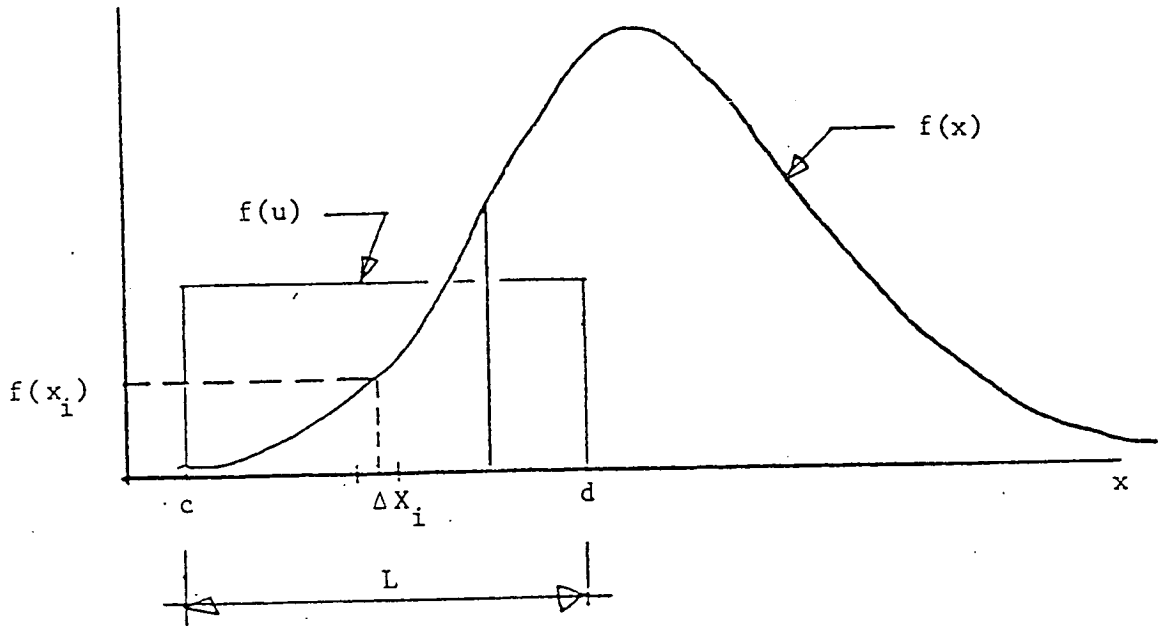


Fig. 4.1 Density functions of X and U

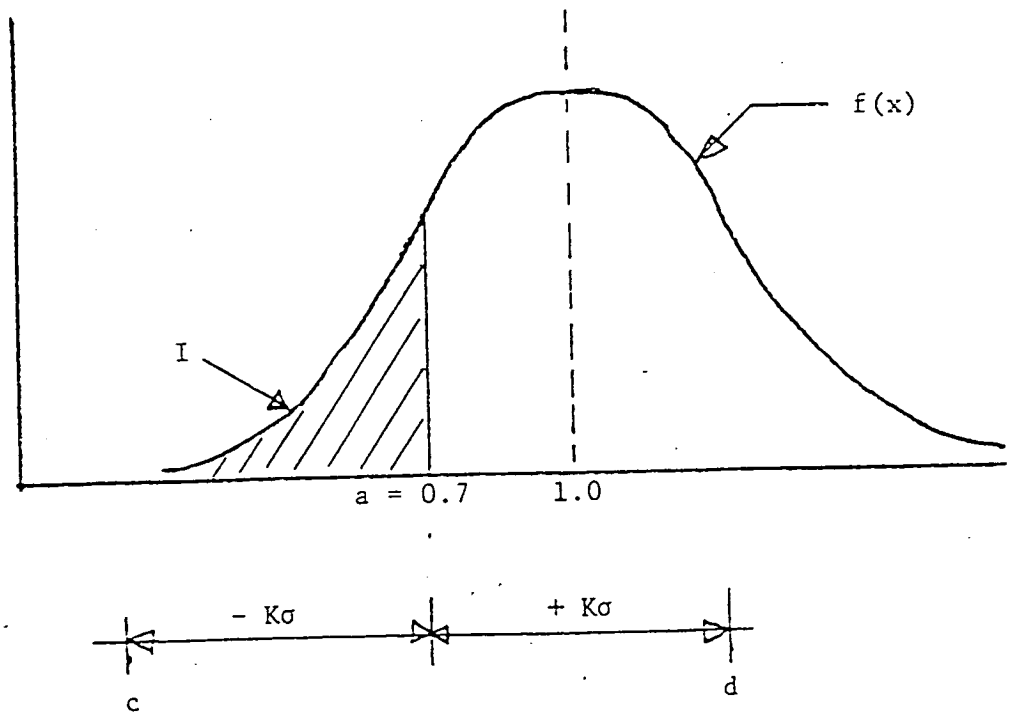


Fig. 4.2 Density function for Example

Consider a random sample of u_i of uniform variates

$$u_i \sim U(c, d) \qquad i = 1, N \tag{4.4}$$

The density function of u is shown in Fig. 4.1.

Let

$$L = d - c \tag{4.5}$$

Note that

$$\Delta x_i \tilde{=} \frac{L}{N} \tag{4.6}$$

And it follows that,

$$I = \sum_{i=1}^{N} \frac{L}{N} \delta_i f(x_i) \tag{4.7}$$

where

$$\delta_i = \begin{cases} 0 & \text{if } u_i > a \\ 1 & \text{if } u_i \leq a \end{cases} \tag{4.8}$$

Example: Compute $P[X \leq 0.70]$ where $X \sim N(1.0, 0.10)$ by Monte Carlo using the mean value scheme. The interval (c, d) is defined as shown in Fig. 4.2. σ is the standard deviation. Here K = 4.5. The reason for not choosing d = a here (which would be more reasonable) is that the scheme of selecting an interval for the integration boundary must be applied in the multidimensional case so it is employed here as well.

First, N was set to 1000. An estimate of I, denoted as $\hat{I}$, was computed. The process was repeated 10 times. Results are given in Table 4.1. Each of the 10 values of $\hat{I}$ are given along with the sample mean and standard deviation of I. The process was repeated for N = 10,000 and the results are given in Table 4.2.

Table 4.1 Monte Carlo Estimate of the Integral of Example

N = 1,000	$\hat{I}$
	1.2274425E-03
	1.1876076E-03
	1.4003464E-03
	1.3360464E-03
	1.1054387E-03
	1.1761633E-03
	1.4718326E-03
	1.4165875E-03
	1.4458908E-03
	1.2413473E-03

Mean of $\hat{I}$ = 1.301E-3

Exact value of I = 1.350E-3

Std. Dev. of $\hat{I}$ = 0.129E-3

Assuming that $\hat{I}$ is normal, 90% confidence intervals on I are estimated as (1.089, 1.513)E-3. Thus, this is the C. I. on I associated with a sample of size 10,000.

Table 4.2 Monte Carlo Estimate of the Integral of Example

N = 10,000	$\hat{I}$
	1.2286508E-03
	1.3768753E-03
	1.3595114E-03
	1.3669265E-03
	1.4250146E-03
	1.2714810E-03
	1.2380191E-03
	1.4196800E-03
	1.3961388E-03
	1.4053878E-03

Mean of $\hat{I}$ = 1.349E-3

Exact value of I = 1.350E-3

Std. Dev. of $\hat{I}$ = 0.075E-3

Assuming that $\hat{I}$ is normal, 90% confidence intervals on I are estimated as (1.226, 1.472)E-3. This is the C. I. on I associated with a sample of 100,000.

4.3 Extension to the Multidimensional Integral

The mean value method can be extended to the multidimensional case. As an example, the two-dimensional problem will be considered because it is easy to describe the problem. Extrapolation of the concepts to higher dimensions is

Shown in Fig. 4.3 is design parameter space for the two random parameter (X, Y). The probability of failure is the volume under the joint pdf in Ω, the failed region. The general strategy for estimating that integral, I, will be as follows.

1. Locate the design point as a reference for the sampling region. Because its "exact" location is not critical, and because computer time is minimized, a crude and fast method (MVFOSM as described in Sec. 4.4) is employed.

2. A sampling region (integration boundary) is defined as shown in Fig. 4.3. The choice of K_1, K_2, K_3, and K_4 is arbitrary. It is important to include all of the probability mass within Ω. As shown in Sec. 4.5, reasonable results are obtained where all $K_i \equiv K = 5$.

3. Uniformly distributed variates u and v are sampled. The distributions are shown in Fig. 4.4 along with the region of integration.

4. g(u, v) is computed to establish whether or not the point lies in Ω.

5. By subdividing region Ω into incremental areas, ΔA_i, the integral I can be approximated as

$$I = \int_{\Omega} f_X \cdot f_Y \, dx \, dy \tilde{=} \sum_{i=1}^{N} f_{Xi} \, f_{Yi} \, \Delta A_i$$

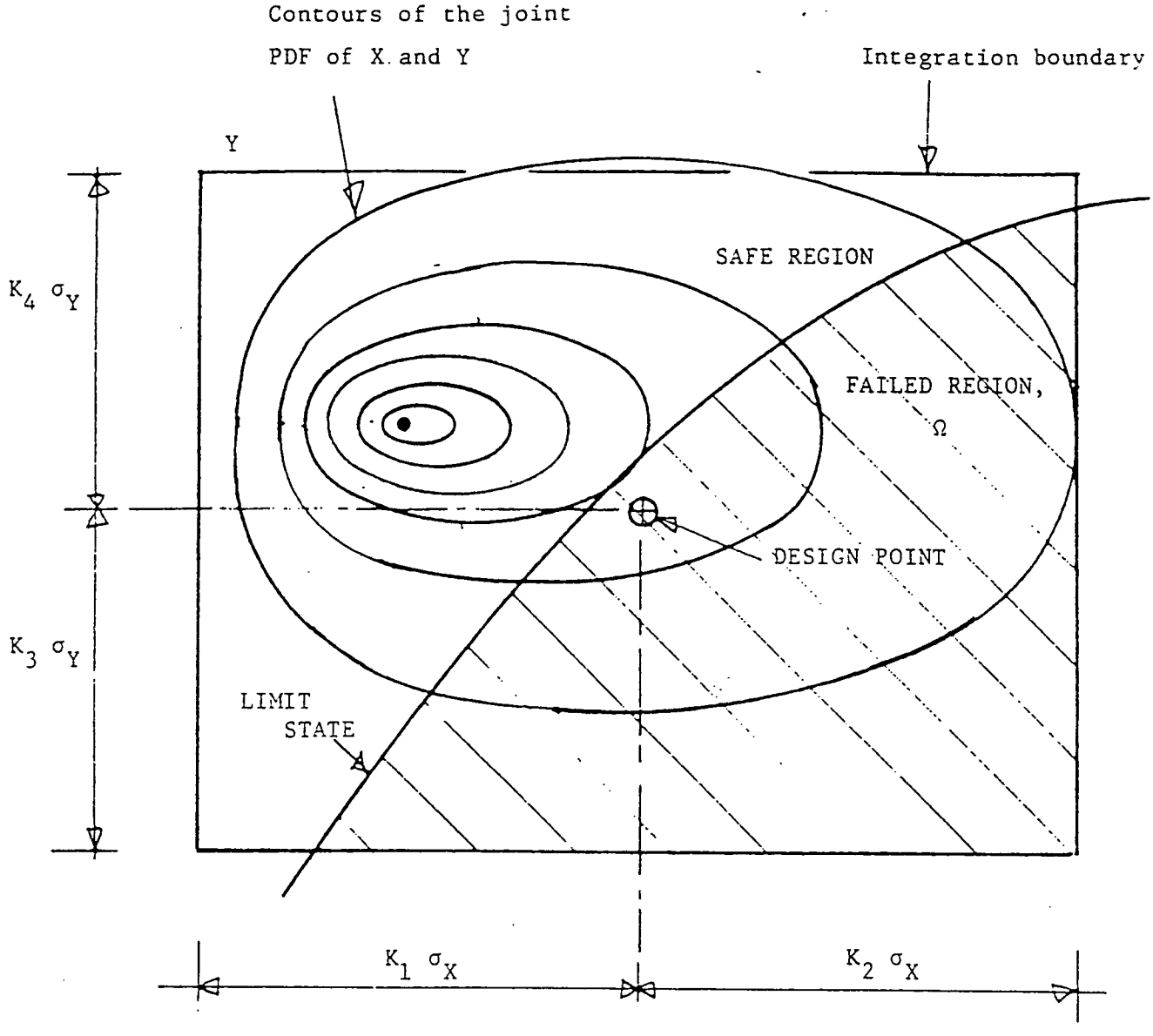


Fig. 4.3 Design parameter space and the region of integration

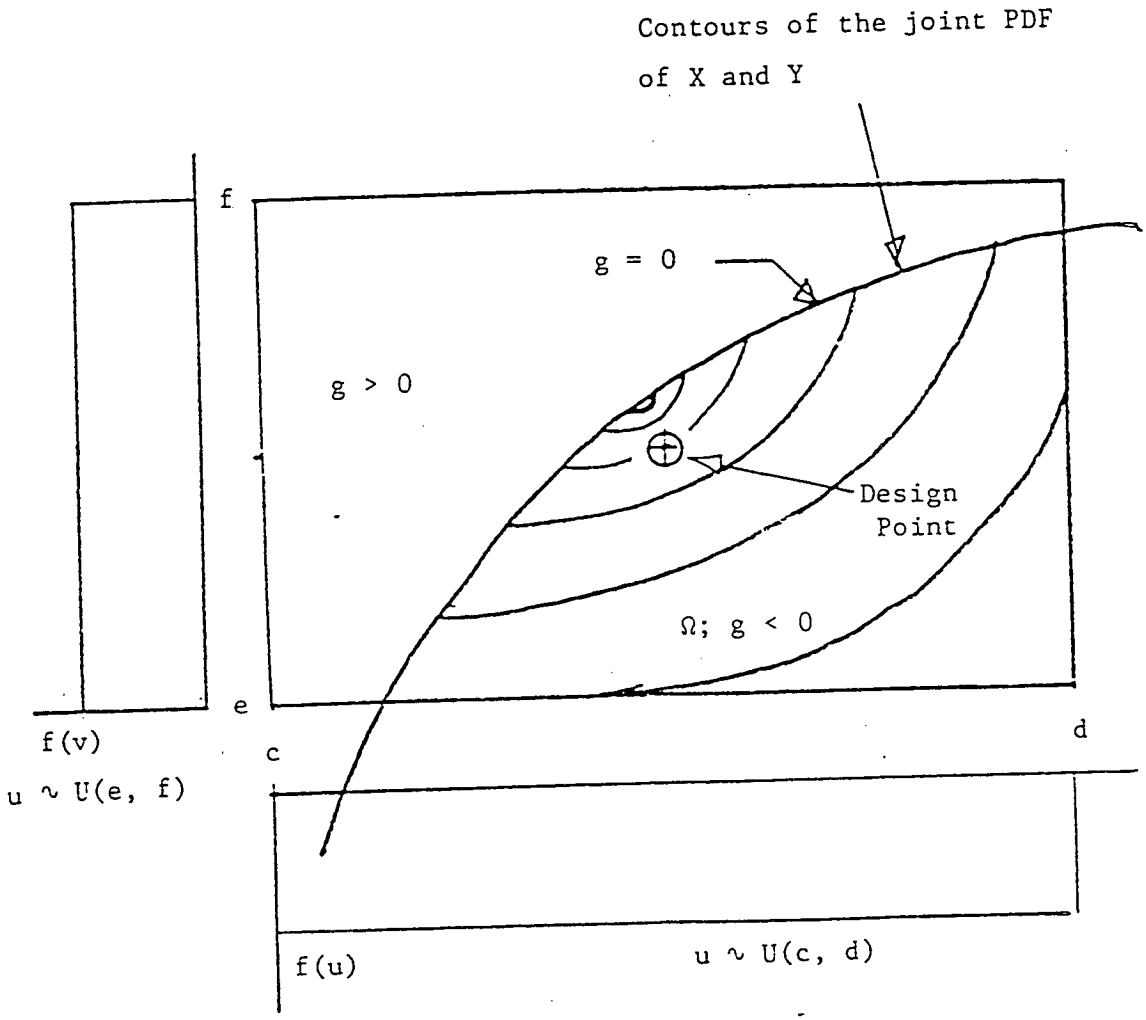


Fig. 4.4 Region of integration and contours of the integrand

where f_{Xi} and f_{Yi} are the pdf's of X and Y respectively at A_i. The integral is estimated by sampling (u_i, v_i); i=1, N, and making the following computation.

$$I \tilde{=} \frac{A}{N} \sum_{i=1}^{N} \delta_i \, f_X(u_i) \, f_Y(v_i) \tag{4.9}$$

where,

$$\delta_i = \begin{cases} 1 & \text{if} \quad g(u_i, v_i) \leq 0 \\ 0 & \text{if} \quad g(u_i, v_i) > 0 \end{cases} \tag{4.10}$$

In this expression for I, ΔA_i is approximated by A/N.

Exactly the same approach is employed for higher dimensional integrals. It can be seen why the Monte Carlo approach is so convenient for evaluation of multidimensional integrals. Employing a straightforward integration scheme, say the trapezoidal rule, computer logic and program statements associated with negotiating the boundaries can become extremely complex. For Monte Carlo, the only operation to define a boundary is the computation of $g(\underset{\sim}{u})$ where $\underset{\sim}{u}$ is the vector of uniform variates.

The "bad news" of Monte Carlo is that large sample sizes are required to reduce confidence intervals on estimates of I to reasonable levels.

4.4 Location of the Design Point

Consider the performance function $g(\underset{\sim}{X})$. It is required to locate the design point (see Figs. 4.3 and 4.4). Note that the design point will depend upon the method used. The Wu/FPI and the Rackwitz-Fiessler methods are expected to produce a "high quality" result. But the Hasofer-Lind method can be employed as well. And a design point can be established using the mean

value first order second moment method (MVFOSM) originally developed for computation of the safety index. This method was found to be fast, but it was later discovered to produce unsatisfactory results for larger problems. The method used in Program SELSAM is the Rackwitz-Fiessler algorithm which was found to produce consistently good results.

4.5 Confidence Intervals: The Efficiency of the Mean Value Method

To run a Monte Carlo integration program, one must first choose (a) the sample size, N, and (b) the region of integration defined by K_1, K_2, K_3, and K_4. For the examples considered in this study, all of the K's were assumed to be the same and equal to K. In all cases, M = 10 repetitions of the evaluation of I were performed for a given N = 10,000 and K. This was done to estimate the distribution of $\hat{I}$ for the purpose of constructing a confidence interval.

To illustrate the results of the analysis, Table 4.3 shows the estimated value of the integral for each of M = 10 repetitions for the first example. N = 10,000 points were used for each estimate $\hat{I}$. Therefore, the sample mean of $\hat{I}$, namely 5.22E-4, is then the best estimate of I and is based on a total sample of 100,000.

The purpose of repeating the integral evaluation (e.g., Table 4.3) is to estimate the variance of the estimator and then construct confidence intervals Consider a vector of estimates of I

$$\hat{I} = (\hat{I}_1, \hat{I}_2, \ldots \hat{I}_M) \tag{4.11}$$

Table 4.3 Example of a Detailed Summary of the Results for a Single Value of K and N.

Performance Function, $g = R - S$

$R \sim LN\ (50., 0.20)$
$S \sim LN\ (20., 0.20)$
} Median and COV

Sample Size, N = 10,000

Region of Integration, K = 5.0

MVFOSM Analyses: $\beta = 2.79$ Design Point

$R^* = 25.14$

$S^* = 24.53$

	$\hat{I}$	$\beta = -\Phi(\hat{I})$	CPU Seconds
1	4.55E-4	3.32	2.25
2	5.22	3.28	2.30
3	4.82	3.30	2.31
4	5.39	3.27	2.27
5	5.08	3.29	2.28
6	5.11	3.28	2.25
7	5.53	3.26	2.28
8	5.09	3.29	2.31
9	5.46	3.27	2.30
10	5.94	3.24	2.33

Total CPU Execution Time = 22.9 seconds

Sample Mean of $\hat{I}$ = 5.22E-4

Exact Value of I = 5.35E-4

Bias = $\frac{\text{Estimated}}{\text{Exact}}$ = 0.98

Sample Standard Deviation of $\hat{I}$ = 0.39E-4

Coefficient of Variation of $\hat{I}$ = 7.3%

The sample mean and variance are,

$$\bar{\hat{I}} = \frac{1}{M} \sum_{i=1}^{M} \hat{I}_i \tag{4.12}$$

$$s_I^2 = \frac{1}{M-1} \sum_{i=1}^{M} (\hat{I}_i - \bar{\hat{I}})^2 \tag{4.13}$$

Let M = 10. The 95% confidence interval on I using an individual estimate is

$$\hat{I} - 2.23\, s_I \leq I \leq \hat{I} + 2.33\, s_I \tag{4.14}$$

The number 2.33 is the student's t variate for n = 10 at a level of 2.5%.

The 95% confidence intervals based on the mean of the estimates is

$$\bar{\hat{I}} - \frac{2.33\, s_I}{\sqrt{10}} \leq I \leq \bar{\hat{I}} + \frac{2.33\, s_I}{\sqrt{10}} \tag{4.15}$$

Example: From the data of Table 4.3, 95% confidence intervals for I are, (in terms of 10^{-4}),

$$5.22 - \frac{2.33(.39)}{3.16} < I < 5.22 + \frac{2.33(.39)}{3.16} \tag{4.16}$$

Or,

$$P = (4.93 < I < 5.51) = 0.95 \tag{4.17}$$

Suppose it is desired to establish the sample size requirement for a given accuracy. For example, find the minimum M to ensure that the value of I will be within $\pm$ 10% of $\hat{I}$ with a confidence of 95%. Assuming that $\hat{I}$ will be normally distributed with a mean of I and standard deviation of $s_I/\sqrt{M}$ it follows that 95% confidence intervals on I are,

$$\left(\hat{I} \left[1 - \frac{1.96\, C_I}{\sqrt{M}} \right] \quad ; \quad \hat{I} \left[1 + \frac{1.96\, C_I}{\sqrt{M}} \right] \right) \tag{4.18}$$

where $C_I = s_I/\hat{I}$. For $\pm$ 10%, let

$$0.10 = \frac{1.96\ C_I}{\sqrt{M}} \tag{4.19}$$

Therefore, the requirement on M is

$$M \geq 384\ C_I^2 \tag{4.20}$$

Unfortunately, one does not know C_I in advance. However, after analyzing several check cases, an approximate relationship between C_I and n is given in Fig. 4.5. These figures are actually more applicable to the stratified sampling version of the mean value method described in Sec. 4.7 below.

Using Fig. 4.5, Eq. 4.20 and Fig. 2.5, one can pre-estimate the sample size requirement and cost. For example, if the response function has 10 random variables, then $C_I = 0.80$ from Fig. 4.5. The number of blocks of 10,000 is given by Eq. 4.20 as M = 246. Thus, the total sample size requirement is 246 x 10^4 or 2.46 million evaluations of the integrand. From Fig. 2.5, the total CPU execution time on the CYBER 175 would be about 344 seconds. This is for 95% confidence for $\hat{I}$ to be within $\pm$ 10% of I.

4.6 Examples of the Mean Value Method

Other examples of the performance of the mean value method are given in Tables 4.4 through 4.7.

The mean value method seems to perform better than the direct (conventional or variance reduction methods. But the literature promises that efficiency of the mean value method will be improved by stratified sampling, i.e., sampling with a higher density of points in the region where $f_{\underset{\sim}{X}}$ is the largest. This is also called group sampling, or selective sampling, and is essentially what is often referred to as "importance sampling." In short, more samples are taken in the more important regions.

Fig. 4.5 C_I As A Function of n For Mean Value Method

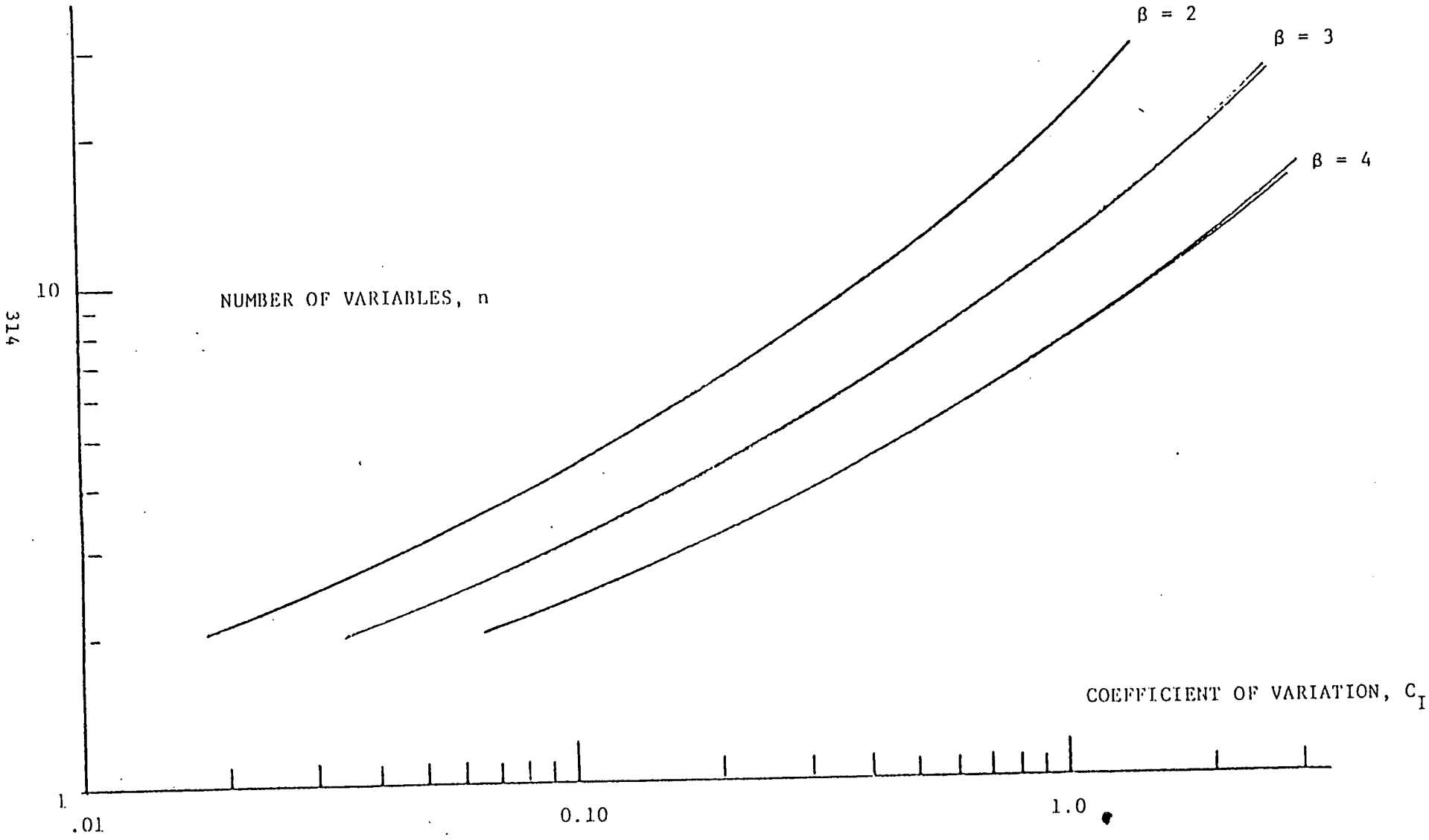


Table 4.4

Example 1; Monte Carlo Integration

Performance Function $g = R - S$

$R \sim LN(\tilde{R}, C_R) = LN(50., 0.20)$

$S \sim LN(\tilde{S}, C_S) = LN(20., 0.20)$

Exact $I \equiv p_f = 5.35E-4$

Sample Size N	Region of Integration K	COV of $\hat{I}$ (%)	Bias*	CPU Execution Time (sec)**
1000	3.5	28	.77	2.5
	4.5	28	.84	2.3
	5.0	30	.85	2.3
5000	5.0	8.3	.94	11.4
10,000	3.5	7.2	.87	24.5
	4.5	7.5	.96	23.3
	5.0	7.5	.98	22.1

*Bias = Estimated I/Exact I; Estimated I is the average of 10 repetitions of $\hat{I}$, each having a sample size of N.

**CYBER 175

Table 4.5

Example 2

Performance Function $g = R - S$

$R \sim WEI\ (20., 2.0)$ Mean and Standard Deviation

$S \sim EVD\ (10., 2.0)$

Exact $I \equiv p_f = 2.86E\text{-}3$

Sample Size N	Region of Integration K	COV of $\hat{I}$ (%)	Bias*	CPU Execution Time (sec)**
1000	3.5	9.29	.88	1.71
	4.5	12.03	.95	1.71
	5.0	13.64	.96	1.71
10,000	3.5	2.17	.91	17.0
	4.5	2.76	.98	17.0
	5.0	3.05	1.00	17.0

*Bias = Estimated I/Exact I

**CYBER 175

Table 4.6

Example 3; Monte Carlo Integration

Performance Function $g = R - S^2$

$R \sim WEI\ (20., 4.0)$

$S \sim FRE\ (3., 0.6)$

Exact $I \equiv p_f = 4.27E\text{-}2$

Sample Size N	Region of Integration K	COV of $\hat{I}$ (%)	Bias*	CPU Execution Time (sec)**
1000	3.5	10.5	.88	2.0
	4.5	11.3	.93	2.0
	5.0	12.6	.93	1.9
10,000	3.5	2.17	.92	20.5
	4.5	2.48	.96	19.4
	5.0	3.02	.98	19.3

*Bias = Estimated I/Exact I

**CYBER 175

Table 4.7

Example 4; Monte Carlo Integration

Performance Function $g = R - \sqrt{300P^2 + 1.92\ T^2}$

$R \sim$ WEI (48., 3.0) Mean and Standard Deviation

$P \sim$ LN (1.0, 0.16)

$T \sim$ EVD (20., 2.0)

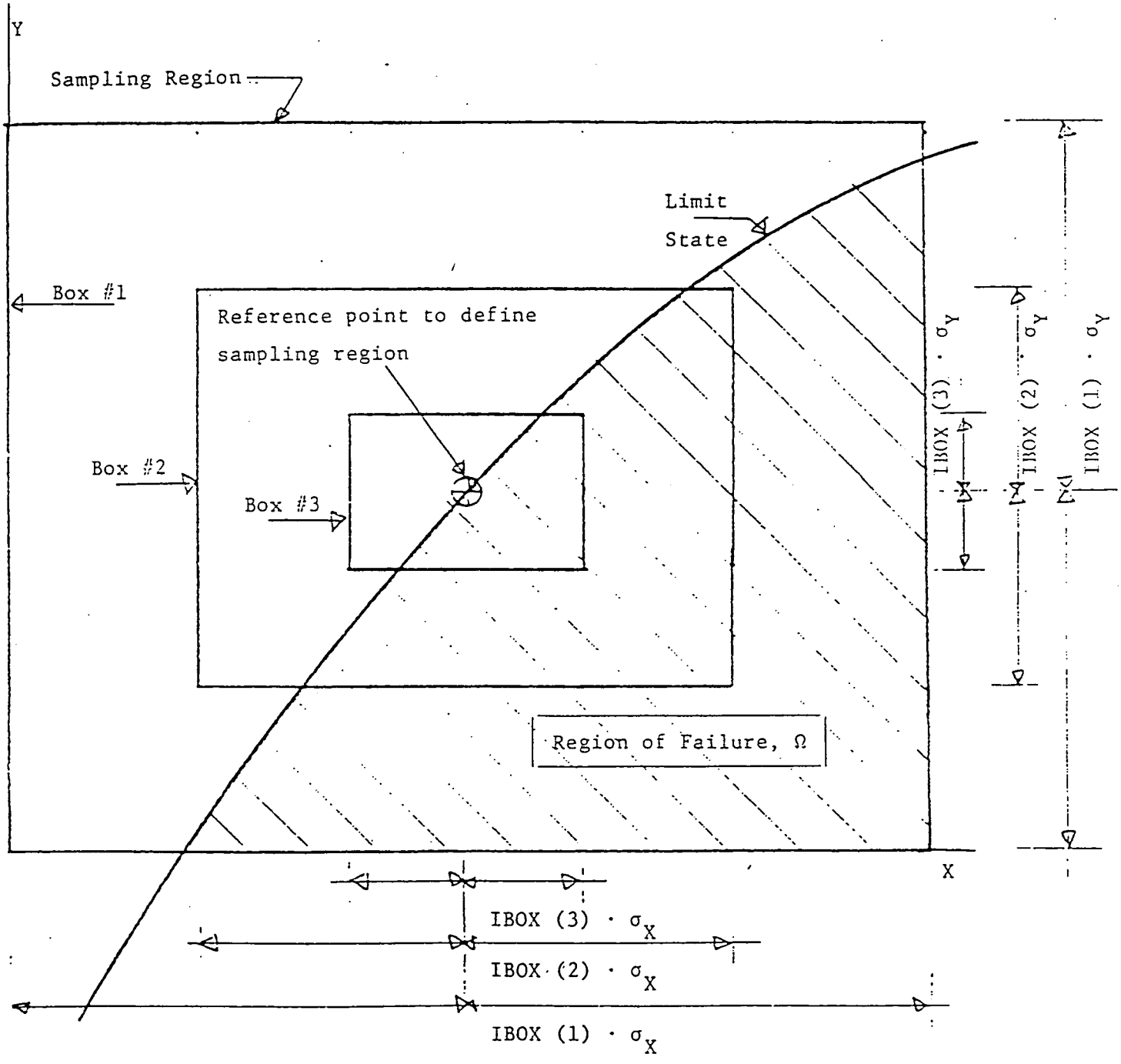
Exact $I \equiv p_f = 1.80E\text{-}3$

Sample Size N	Region of Integration K	COV of $\hat{I}$ (%)	Bias*	CPU Execution Time (sec)**
1000	3.5	5.3	.86	3.4
	4.5	9.1	.94	3.3
	5.0	10.1	.96	3.3
5000	5.0	6.5	1.00	16.4
10,000	3.5	3.7	.88	32.2
	4.5	4.8	.97	32.4
	5.0	5.1	.99	32.5

*Bias = Estimated I/Exact I

**CYBER 175

Fig. 4.6


Y
Sampling Region
Limit
State
Box #1
Reference point to define
sampling region
Box #2
Box #3
Region of Failure, Ω
X
IBOX (3) · σY
IBOX (2) · σY
IBOX (1) · σY
IBOX (3) · σX
IBOX (2) · σX
IBOX (1) · σX

4.7 Stratified Sampling; Extension of the Mean Value Method

Fig. 4.3 illustrated for two dimensions, the joint probability density function (pdf), the limit state, and the region of failure, Ω. Shown is a "reference point," (in this case a design point which could be obtained by MVFOSM or any of the fast probability integration methods) which is "close" to the peak of the pdf. This reference point is used to define the sampling region.

A summary of the stratified sampling scheme is shown in Fig. 4.6. First, the reference point is established. In program SELSAM, it is defined by a Hasofer-Lind or Rackwitz-Fiessler design point (user's option). Then the user must decide

1. The number of boxes
2. The size of each box
3. The total number of points; i.e., the sample size
4. The number of points in each box, i.e., how the sample is stratified

Because $f_{\underset{\sim}{X}}$ will have its peak close to the reference point, it is anticip that the density of points in Box #3 should be high. Fewer points should be in Box #2 and still fewer in Box #1. Then, if the user wants to estimate confidence intervals on his point probability estimate, the sample should be repeated.

Studies on how to select the parameters above to minimize the sample size for a given confidence interval have been inconclusive. But for some sample problems, the parameters as given in Table 4.8 have performed well.

4.8 Program SELSAM

Program SELSAM performs, by Monte Carlo, numerical integration of the probability of failure integral using stratified sampling. When only one box is chosen, (no stratified sampling) the program algorithm is the mean value method.

A listing of Program SELSAM is given in Appendix D.

Examples of Program SELSAM are given in Tables 4.9 and 4.10. The example of Table 4.9 has only one box and is therefore the mean value method. In the second example of Table 4.10, the formula for defining stratified sampling as given by Table 4.8 was employed.

The program has been exercised on several example problems. The performance of the program is measured by its accuracy in making point probability calculation and its corresponding CPU execution time.

TABLE 4.8 Preliminary Recommendaion for Defining the Parameters for Stratified Sampling

• Number of Boxes	4
• Size of Each Box Number of standard deviations ± from the reference point	
Box 1	5
Box 2	3
Box 3	2
Box 4	1
• Total Number of Points in one sample	10,000
• Points in Each Box	
Box 1	500
Box 2	1000
Box 3	2000
Box 4	6500
• Number of Samples (Note that this is the value which is being used for the purpose of estimating confidence intervals associated with the sampling.)	10

TABLE 4.9

EXAMPLE 1: Evaluation of the Probability Integral by the Mean Value Method

(Sampling in only one block)

Response function:

$$g = R - \sqrt{300P^2 - 1.92T^2}$$

		Mean	Std. Dev.
R	Weibull	48.	3.
P	lognormal	1.0*	0.16*
T	EVD	20.	2.

*The median and COV are $\tilde{P} = 0.9874$ and $C_P = 0.16$

- Use only one box, ISTRIP = 1
- Box (sampling region) is $\pm$ 5σ in all directions
- Take IBOX = 10,000 points
- Repeat process NT $\equiv$ K = 10 times.

This is the input for this problem.

```
0,10,3
1
5.,10000
R
1,48.,3.
P
4,.987440632,.16
T
3,20.,2.
0.
```

The output is on the next page.

VARIABLE	DIST.	MEAN	STD. DEV
R	WEIBULL	4.8000E+01	3.0000E+00
P	LOG N.	9.8744E-01	1.6000E-01
T	EVD	2.0000E+01	2.0000E+00

K = 10

SAMPLE (POINTS) = 10000

STRIP (SIGMA) = 5.00

INITIAL STARTING POINT (REDUCED VARIATES)
-2.564 1.783 1.945

ESTIMATE OF I	BETA	CPU SEC
1.6759E-03	2.933	3.03
1.7456E-03	2.921	3.04
1.9782E-03	2.882	3.04
1.7872E-03	2.913	3.05
1.6600E-03	2.936	2.95
1.8472E-03	2.903	3.02
1.8751E-03	2.898	2.98
1.7543E-03	2.919	3.04
1.7949E-03	2.912	3.03
1.6824E-03	2.932	3.08

AVG. OF ESTIMATION = 1.7801E-03

STANDARD DEVIATION = 9.9832E-05

TABLE 4.10

EXAMPLE 2: Evaluation of the Probability Integral by Stratified Sampling

(An extension of the mean value method)

Response function:

$$g = R - \sqrt{300P^2 - 1.92T^2}$$

		Mean	Std. Dev.
R	Weibull	48.	3.
P	lognormal	1.0*	0.16*
T	EVD	20.	2.

*The median and COV are $\tilde{P} = 0.9874$ and $C_P = 0.16$

- Use four boxes, ISTRIP = 4
- Boxes are respectively ($\pm$ 5, $\pm$ 3, $\pm$ 2, $\pm$ 1) σ in all directions
- Samples in each box are respectively (500, 1000, 2000, 6500)
- Repeat process NT $\equiv$ K = 10 times

This is the input for this problem.

```
0,10,3
4
5.,500
3.,1000
2.,2000
1.,6500
R
1,48.,3.
P
4,.987440632,.16
T
3,20.,2.
0.
```

The output is on the next page.

VARIABLE	DIST.	MEAN	STD. DEV
R	WEIBULL	4.8000E+01	3.0000E+00
P	LOG N.	9.8744E-01	1.6000E-01
T	EVD	2.0000E+01	2.0000E+00

K = 10

SAMPLE (POINTS) = 500 1000 2000 6500

STRIP (SIGMA) = 5.00 3.00 2.00 1.00

INITIAL STARTING POINT (REDUCED VARIATES)
-2.564 1.783 1.945

ESTIMATE OF I	BETA	CPU SEC
1.6086E-03	2.946	3.14
1.8204E-03	2.908	3.11
1.9675E-03	2.883	3.10
1.7617E-03	2.918	3.11
1.7030E-03	2.929	3.09
1.7820E-03	2.914	3.09
1.8696E-03	2.899	3.08
1.6840E-03	2.932	3.12
1.7000E-03	2.929	3.12
1.8449E-03	2.904	3.12

AVG. OF ESTIMATION = 1.7742E-03

STANDARD DEVIATION = 1.0538E-04

5.0 THE HARBITZ ALGORITHM

5.1 Preliminary Remarks

In a 1986 issue of Structural Safety, Alf Harbitz presented a Monte Carlo method which estimates point probabilities [8]. The algorithm is presented as an "efficient" method. The decision was made by the UA team to develop the method and compare its performance to other available Monte Carlo schemes.

The performance function is given as $g(\underset{\sim}{X})$ where $\underset{\sim}{X}$ is a vector of random variables. The goal of analysis is to compute

$$p = P[g(\underset{\sim}{X}) \leq 0] \tag{5.1}$$

Consider $\underset{\sim}{X}$ as a two-dimensional vector. Fig. 5.1 shows the region where $g \leq 0$. From probability theory, p can be evaluated by,

$$p = \int_{\Omega} f_{\underset{\sim}{X}}(\underset{\sim}{x})\, d\underset{\sim}{x} \tag{5.2}$$

But the integral is difficult to evaluate for higher order vectors. The Harbitz method provides an estimate of p.

While the method is described in detail in Ref. 8 , a summary is provided as follows. Also described are modifications to the method to improve its perfromance relative to the original Harbitz algorithm.

5.2 Expression for Point Probability

The basic variables $\underset{\sim}{X}$ can be transformed to standard normal variates $\underset{\sim}{x}$ using the relationship,

$$F_i(X_i) = \Phi(x_i) \tag{5.3}$$

The lower case x_i denotes the transformed variables. Φ is the standard normal distribution function (cdf) and F_i is the cdf of X_i. Using the transformation, the performance function can be written in terms of x_i. This function $g_1(\underset{\sim}{x})$ when set equal to zero becomes the limit state. Fig. 5.2 shows the region where $g_1 \leq 0$ in the space of standard normal variates.

The minimum distance from the origin to g_1 is given as β. In a first order reliability method (FORM), β provides a first approximation to p, i.e.,

$$p = \Phi(-\beta). \tag{5.4}$$

But an exact expression can be formulated for p. Note that because $\underset{\sim}{x}$ is standard normal, the probability that $\underset{\sim}{x}$ will be in the "β-sphere will be

$$P[|\underset{\sim}{x}| \leq \beta] = P[|\underset{\sim}{x}|^2 \leq \beta^2] = \Gamma_n(\beta^2) \tag{5.5}$$

where Γ_n is the chi-square cdf with n degrees of freedom. n is the size of the vector $\underset{\sim}{X}$. Harbitz uses this fact and shows, using elementary probability operations, that

$$\begin{aligned} p &= P[g \leq 0] \\ &= P[g < 0 \,\big|\, |\underset{\sim}{x}| > \beta] \cdot (1 - \Gamma_n(\beta^2)) \end{aligned} \tag{5.6}$$

Evaluation of p requires application of a combination of reliability methods.

5.3 Computation of p; Basic Considerations

Numerical FORM can be employed to compute β. One method is the Rackwitz-Fiessler algorithm. The second term of Eq. 5.6 is easily calculated

Monte Carlo is used to estimate the first term of Eq. 5.6. For convenience, we will let this probability be denoted as p_1. The scheme for extimati p_1 is as follows:

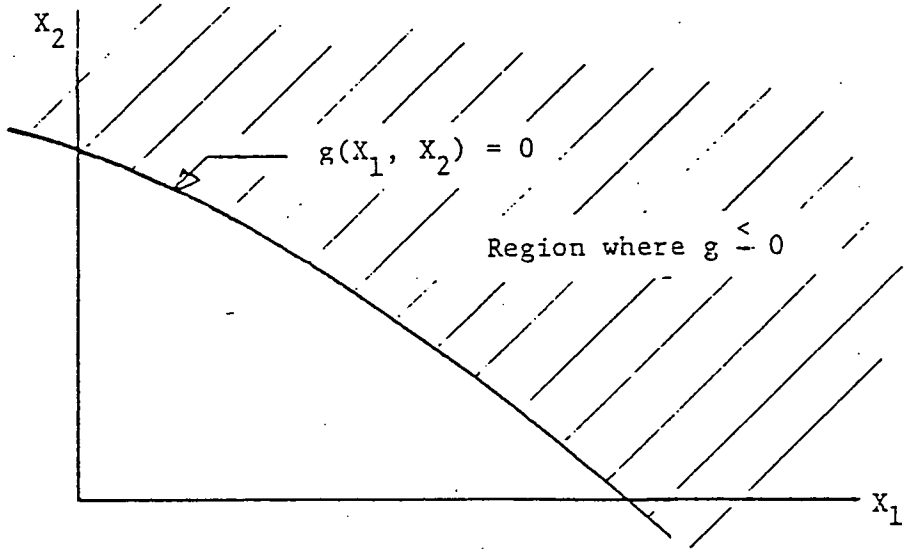


Fig. 5.1 Region where $g \leq 0$

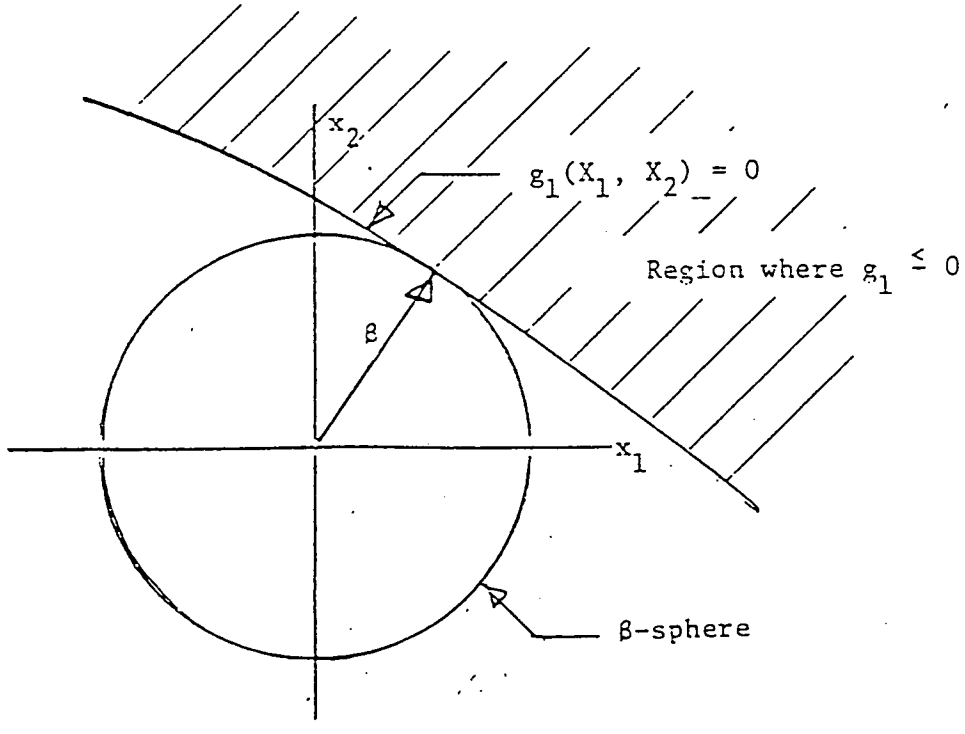


Fig. 5.2 Region Where $g_1 \leq 0$

1. Sample $\underset{\sim}{x}$ outside the β-sphere
2. Transform to basic variables, $\underset{\sim}{X}$
3. Compute $g(\underset{\sim}{X})$
4. Repeat process K times

Then

$$p_1 = P[g < 0 \,\big|\, |\underset{\sim}{x}| > \beta] \overset{\sim}{=} \hat{p}_1 \tag{5.7}$$

where,

$$\hat{p}_1 = \frac{K_{(-)}}{K} \tag{5.8}$$

and where $K_{(-)}$ is the number of samples for which $g < 0$. $\hat{p}_1$ is the extimator of p_1.

5.4 A Note About Efficiency

The reason why this method promises to be efficient is illustrated in Fig. 5.3. The random points outside the β-sphere are as illustrated. Note that a relatively high percentage will fall in the region where $g \overset{<}{\sim} 0$. In practice $\hat{p}_1$ will typically fall in the range (0.05, 0.25). For the special case where g is linear in normal $\underset{\sim}{X}$, g_1 will be a straight line and $p_1 \overset{\sim}{=} 0.10$ The point is that confidence intervals on p_1, for a given K, are relatively large if p_1 is very small. But if $\hat{p}_1$ is in the range as indicated, narrow condifence intervals can be obtained with relatively small sample sizes, K.

Example . . . For large K, p_1 will be approximately normal, and it can be shown that,

$$P[\hat{p}_1(1 - \gamma) \overset{<}{\sim} p_1 \overset{<}{\sim} p_1(1 + \gamma)] \overset{\sim}{=} 1 - \alpha \tag{5.9}$$

where,

$$\gamma = z_{\alpha/2} \sqrt{\frac{1 - \hat{p}_1}{\hat{p}_1 K}} \tag{5.10}$$

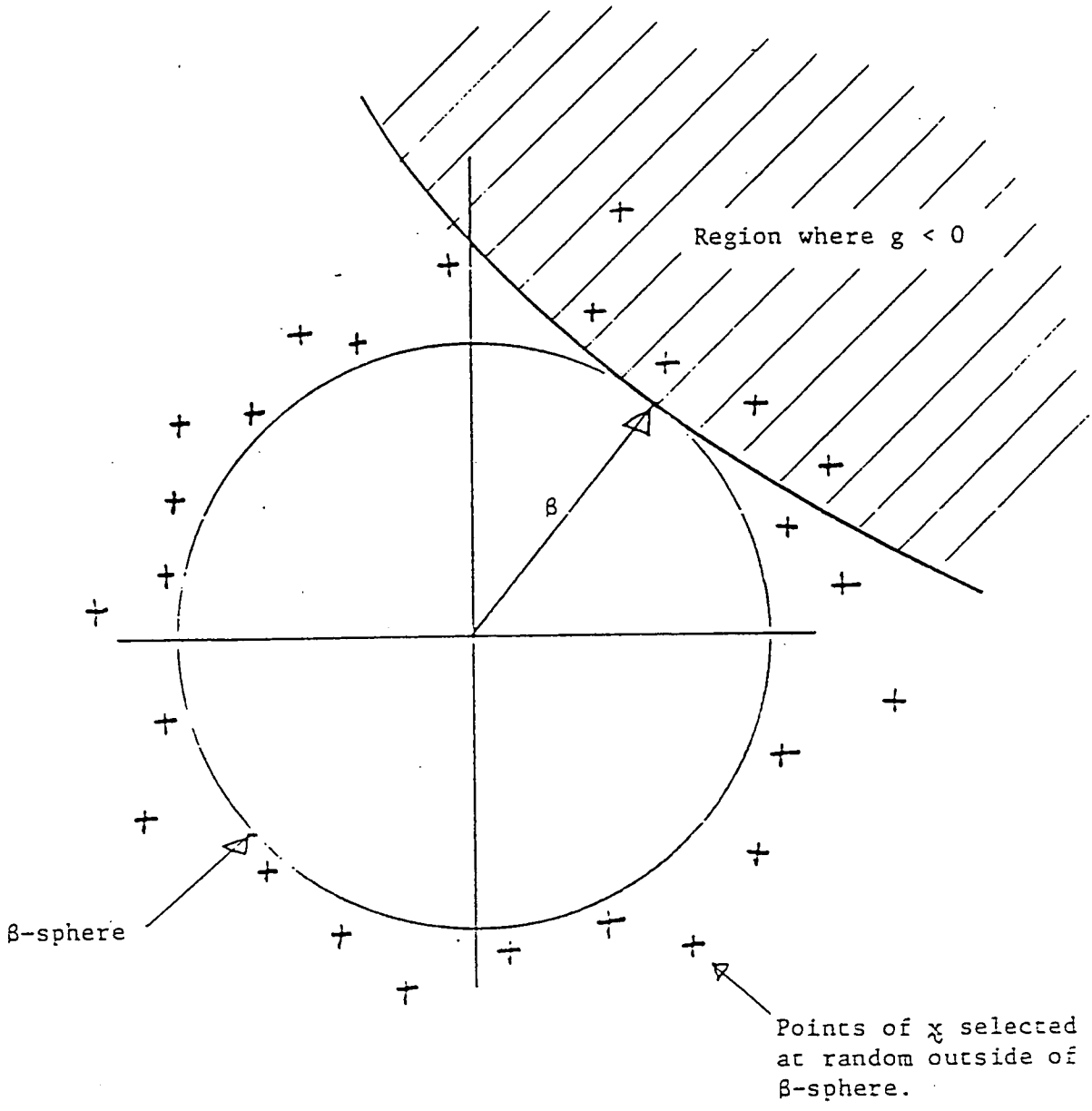


Fig. 5.3 Example of Sampling Outside the β-sphere (2-D representation)

Compute the required K for 95% confidence intervals within $\pm$ 10% of $\hat{p}_1$.

$$\gamma = 0.10 \qquad \text{and} \qquad z_{\alpha/2} = 1.96$$

Then,

$$K = \frac{384\ (1 - \hat{p}_1)}{\hat{p}} \tag{5.11}$$

For $\hat{p}_1 = 0.10$, $K \tilde{=} 3500$. For $p_1 = 0.001$, $K \tilde{=} 384{,}000$.

This exercise clearly demonstrates why it is efficient to formulate the problem so as to avoid low probability levels.

5.5 How to Sample $\underset{\sim}{x}$ Outside the β-Sphere $|\underset{\sim}{x}| > \beta$

Harbitz proposes the following as an efficient sampling procedure. The first step is to transform the standard normal variates $\underset{\sim}{x}$ to polar coordinate

$$\underset{\sim}{x} \rightarrow (R, \underset{\sim}{\Theta}) \tag{5.12}$$

where $\underset{\sim}{\Theta} = (\Theta_1, \Theta_2, \ldots \Theta_{n-1})$ defines the direction of $\underset{\sim}{x}$ and R defines the length of $\underset{\sim}{x}$. R and $\underset{\sim}{\Theta}$ are independent. Now it is required to obtain a random sample of R and $\underset{\sim}{\Theta}$.

R^2 will have an χ^2 distribution with n degrees of freedom.

$$f_{R^2}(r^2) = \frac{(r^2)^{(n/2)-1} \exp\left(-\frac{1}{2} r^2\right)}{2^{n/2}\ \Gamma(n/2)} \tag{5.13}$$

where $\Gamma(\cdot)$ is the gamma function. R can be sampled from this distribution. $\underset{\sim}{\Theta}$ is sampled as described below. The jth random vector of $\underset{\sim}{x}$ can be sampled as

$$\underset{\sim}{x}_j = \underset{\sim}{\zeta}_i\ R_i \tag{5.14}$$

where,

$$\underset{\sim}{\zeta}_i = \frac{\underset{\sim}{Y}_i}{|\underset{\sim}{Y}|_i} \tag{5.15}$$

and where $\underset{\sim}{Y}_i$ is the ith sample of a vector of $Y_j \sim N(0, 1)$, $j = 1, n$. Thus, $\underset{\sim}{\zeta}_i$ is a random direction unit vector in $\underset{\sim}{x}$ space. This corresponds to sampling a random $\underset{\sim}{\theta}$.

But we want to sample so that $\underset{\sim}{x}_j$ lies outside the β-sphere, i.e., $R > \beta$. The well known "rejection technique" will be employed [8]. See Fig. 5.4.

Define the sampling domain as $[r_1, r_2]$ where $r_1 = \beta$ and $r_2 > \beta$. Experience has shown that accurate results are obtained when $r_2 > 3 + \beta$. To improve the efficiency, perform a transformation

$$U = \exp(-\frac{1}{\alpha} R^2) \tag{5.16}$$

where $\alpha \geq 2$ is a constant whose optimal value depends upon β (see Sec. 5.6). The sampling domain for U then is,

$$[u_1, u_2] = \left[\exp\left(-\frac{1}{\alpha} r_2^2\right), \exp\left(-\frac{1}{\alpha} r_1^2\right)\right] \tag{5.17}$$

Given that R^2 is $\chi^2(n)$, the density function of u, denoted as h(u) is proportional to,

$$h(u) = [-\ln (u)]^{(n/2)-1} u^{(\alpha/2)-1} \tag{5.18}$$

A typical function h(u) is plotted in Fig. 5.5. Comparison of Fig. 5.4 and 5.5 provides the motivation for making the transformation of Eq. 5.16.

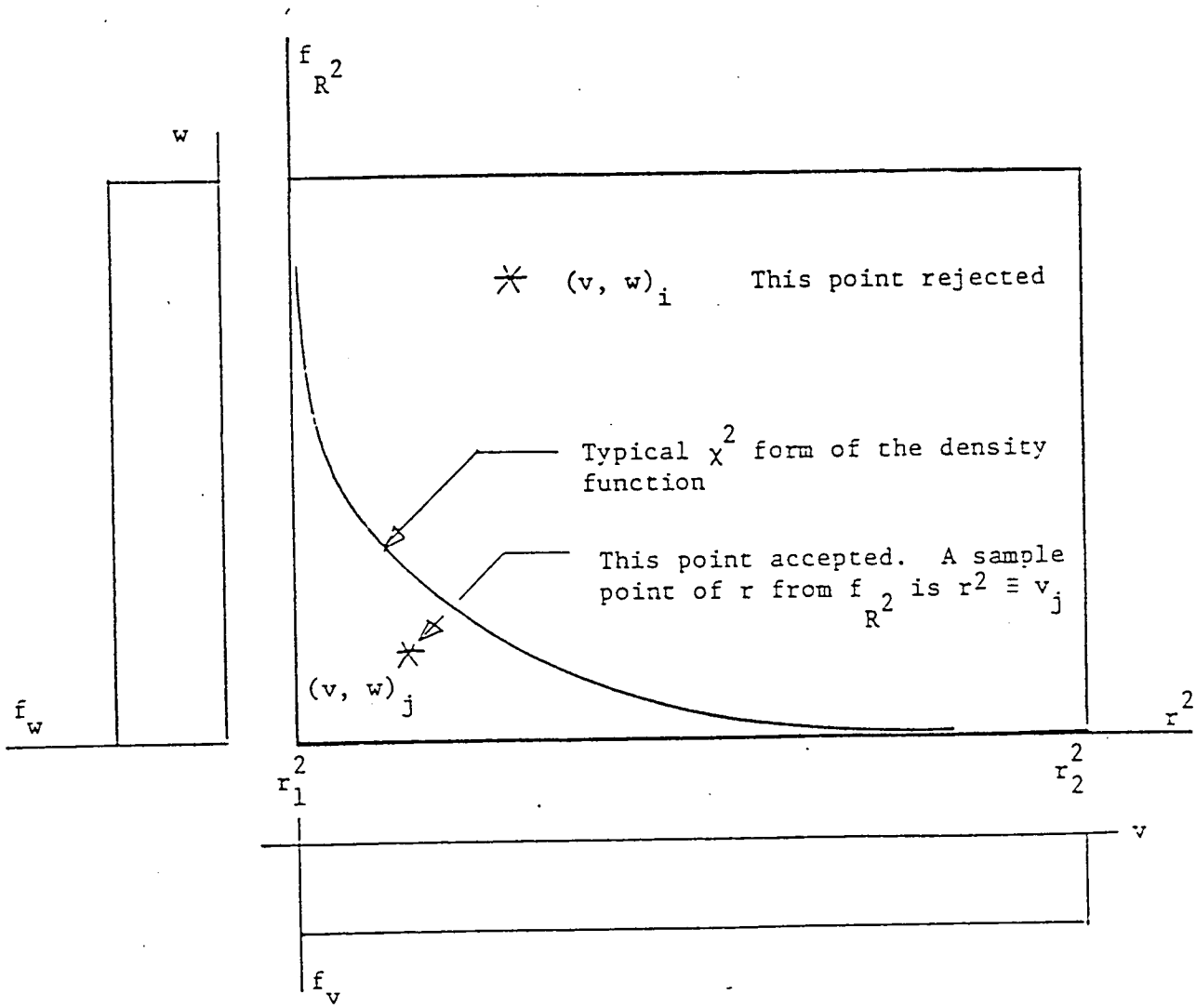
The process of sampling points, as demonstrated in Fig. 5.5, continues until we have u_i; $i = 1, K$. Then the sample of R_i is obtained from Eq. 5.16.

$$R_j = \sqrt{-\alpha \ln u_j} \qquad j = 1, K \tag{5.19}$$

5.7 How to Find α

The one detail missing from the above discussion is how to specify α. We would like to select α so that the rejection area, as illustrated in Fig. 5.5 is minimized. S. J. Lee has developed a simple program which, for a

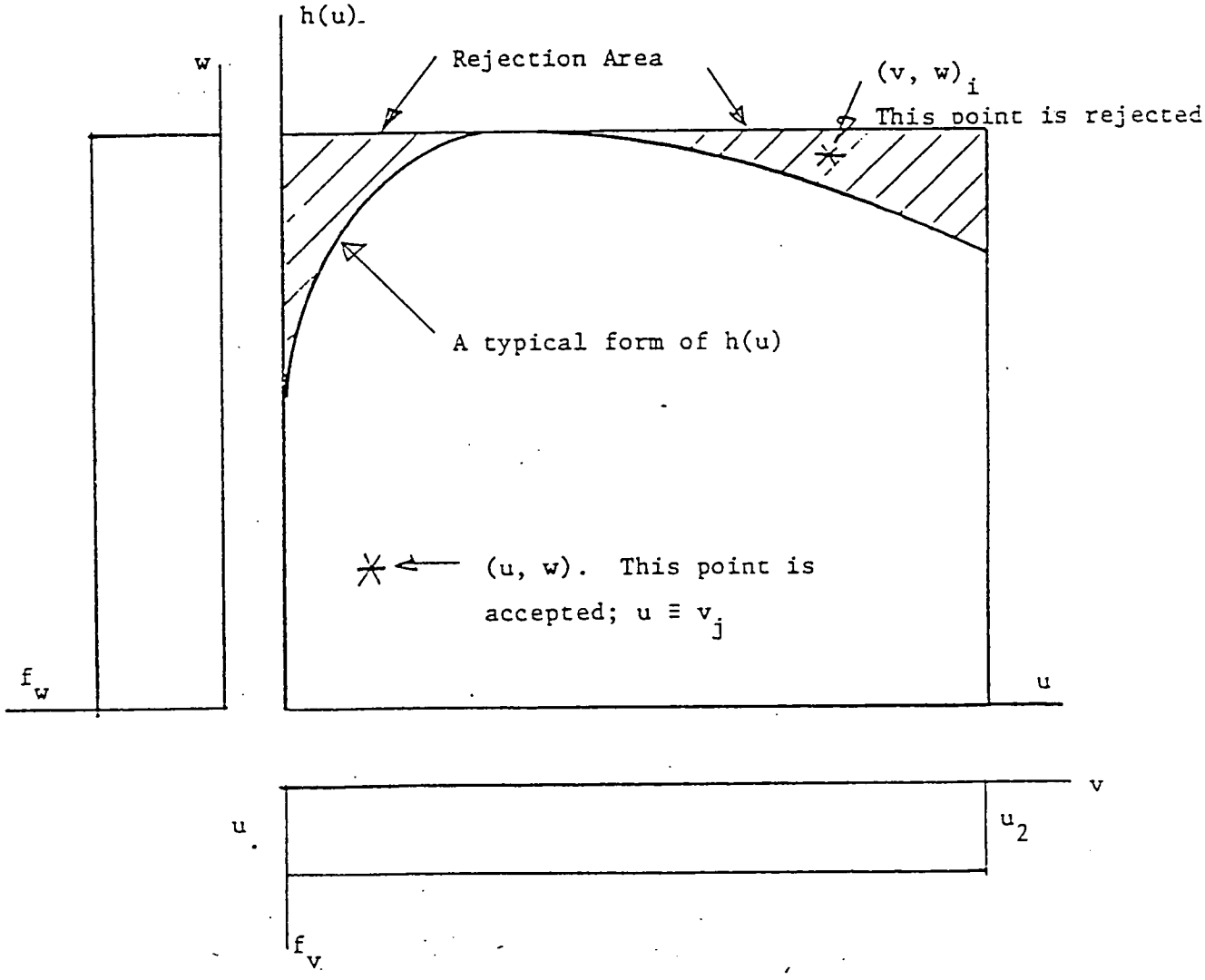
In the rejection method, a pair of uniformly distributed points (v, w) are sampled as shown.



Note that for the form of the distribution, many points are rejected.

Fig. 5.4 The Rejection Technique

Pairs of uniformly distributed variates (v, w) are sampled as shown.



This shows why the transformation of Eq. 5.16 was made. For this form of h(v), very few points are rejected.

Fig. 5.5 How U_i is Sampled

given u_1, u_2, and n, minimizes the rejection area. This is a feature in the UA/Harbitz program.

5.7 The Estimate of p

Finally, a sample of $\underset{\sim}{x}_j$; j = 1, K can be made from Eq. 5.14. As indicated in Sec. 5.3, from $\underset{\sim}{x}_j$, $\underset{\sim}{X}$ is computed, . . . then $g(\underset{\sim}{X})$. Finally, $\hat{p}_1$ is computed by Eq. 5.8, and the point probability estimate is

$$\hat{p} = \hat{p}_1 \left[1 - \Gamma_n (\beta^2)\right] \tag{5.20}$$

5.8 The UA/Harbitz Program

The listing of the UA/Harbitz program for computing point probabilities is given in Appendix F along with a description of the input.

An example is presented in Tables 5.1 and 5.2. A definition of the problem and an example of the input file is provided in Table 5.1. Attached t the output, given in Table 5.2, are notes which describe some of the terms.

5.9 Efficiency of the Harbitz Method

The Harbitz algorithm for point probability estimation by Monte Carlo looks promixing as an independent check on FPI in NESSUS. But the efficiency of the Harbitz method depends strongly upon the number of independent random design factors as well as the probability level. Harbitz efficiency decreases substantially with increasing numbers of random variables in the response function. Also, efficiency decreases as the probability levels become lower, but the loss of efficiency is far less than with conventional Monte Carlo. In the following, the efficiency of the Harbitz method is quantified and compared to direct Monte Carlo.

Table 5.1 Example Problem for Harbitz Method

$$g(\underset{\sim}{X}) = R - \sqrt{300P^2 + 1.92T^2}$$

$\mu_R = 48$	$\mu_P = 1.0$	$\mu_T = 20$
$\sigma_R = 3$	$\sigma_P = 0.16$	$\sigma_T = 2$
$R \sim$ NEI	$P \sim$ LN	$T \sim$ EVD

This is an example of the input file

```
THIS IS EXAMPLE 7
1.D-4,3,10000,0.
R
1.,48.,3.
P
4.,.987440632,.16 *
T
3.,20.,2.
```

*Note that P is lognormal; thus the median

$$\tilde{P} = \mu_P / \sqrt{1 + C_P^2}$$

is entered.

Table 5.2 Output of Harbitz Program For Example Problem

DESIGN VARIABLES

VARIABLE	DISTRIBUTION	MEAN/MEDIAN	STD/COV
R	WEIBULL	4.8000E+01	3.0000E+00
P	LOG	9.8744E-01	1.6000E-01
T	EVD	2.0000E+01	2.0000E+00

(NOTE: THE MEDIAN AND COV USED FOR LN)

BETA (SPHERE) = 3.085 ⟶ First order reliability analysis (R-F)

NUMBER OF VARIABLES = 3

AREA RATIO, AR = .9934 ⟶ 99.34% of points sampled will be accepted

ALPHA = 2.1880 ⟶ value of α corresponding to the area ratio

NUMBER OF SAMPLES = 10000 ⟶ value of K

TOTAL NUMBER OF G < 0 = 782 ⟶ value of $K_{(-)}$

TOTAL NUMER OF POINTS SAMPLED = 10089 ⟶ Actually had to sample 10,089 poi 89 were rejected.

PROBABILITY OUTSIDE BETA SPHERE = 2.318E-2 ⟶ $1 - \Gamma_n(\beta^2)$

PROBABILITY OF FAILURE = 1.81274E-03 ⟶ The central result; Eq. 5.20

BETA = 2.90905

95 CONFIDENCE INTERVAL ON PF

LOWER = 1.69075E-03

UPPER = 1.93472E-03

} Eq. 5.9

CPU EXECUTION TIME (SEC.) = 4.17

Consider the response function $Z = Z(\underset{\sim}{X})$ where $\underset{\sim}{X}$ is a vector of n independent random variables $(X_1, \ldots X_n)$. To evaluate the CDF of Z at point Z_o let

$$g(\underset{\sim}{X}) = Z(X) - Z_o \tag{5.21}$$

The CDF of Z at Z_o is

$$F(Z_o) = P[Z(\underset{\sim}{X}) \leq Z_o] \tag{5.22}$$

$$= P[g(\underset{\sim}{X}) \leq 0]$$

By conventional or direct Monte Carlo, a random sample of size K is obtained, and the CDF of Z at Z_o is estimated as

$$\hat{p}_1 = \frac{K_{(-)}}{K} \tag{5.23}$$

where $K_{(-)}$ is the number of samples for which $g \leq 0$. Thus, p_1 is an estimate of $p = F(Z_o)$.

$1 - \alpha$ confidence intervals on $p = F(Z_o)$ are given (for large K) as,

$$P[\hat{p}_1 (1 - \gamma) \leq \hat{p} \leq p_1 (1 + \gamma)] = 1 - \alpha \tag{5.24}$$

where

$$\gamma = z_{\alpha/2} \sqrt{\frac{(1-\hat{p}_1)}{\hat{p}_1 K}} \tag{5.25}$$

The efficiency of the method is described by the number of samples (K) required so that $\hat{p}_1$ is within $\pm$ 10% of $F(Z_o)$ with a confidence of 95%. Thus,

$$\gamma = 0.10 \qquad z_{\alpha/2} = 1.96$$

And the relationship of K with $\hat{p}_1$ is given as

$$K = \frac{384 \ (1 - \hat{p}_1)}{\hat{p}_1} \tag{5.26}$$

Eq. 5.26 is shown in Fig. 5.6 as the R = 1 curve. Clearly efficiency i very poor at lower probability levels.

To get an idea of sampling costs, the following approximations were observed on the UA CYBER 175.

CPU Execution Time, (seconds)

$$T = (1.4E\text{-}4)\ K \cdot n \tag{5.27}$$

K = sample size

n = number of variables

Computer charges at \$130/hr; cost C in \$)

$$C = 0.036T \tag{5.28}$$

For example, a response function having n = 5 variables, requiring a sample of $K = 10^5$, would run for approximately T = 70 seconds (1.2 minutes) and cost C = \$2.50 on the CYBER 175.

The Harbitz method is a scheme of selective sampling (coule be considere as a form of importance sampling). The point estimate of $p = F(Z_o)$ can be written form Eq. 5.20 as

$$\hat{p} = \hat{p}_1 R \tag{5.29}$$

where R is the reduction factor, defined as

$$R = 1 - \Gamma(\beta^2;\ n) \tag{5.30}$$

$$0 < R \leq 1$$

R is the probability that $\underset{\sim}{X}$ sampled at random will fall outside a sphere in n-dimensional u-space (space of transformed standard normals) of radius β. Plotted in Fig. 5.7 is R as a function of n and β.

The value of the Harbitz method can be seen upon considering Eq. 5.29 and Fig. 5.6. Clearly as R gets smaller, the sample size K required becomes small

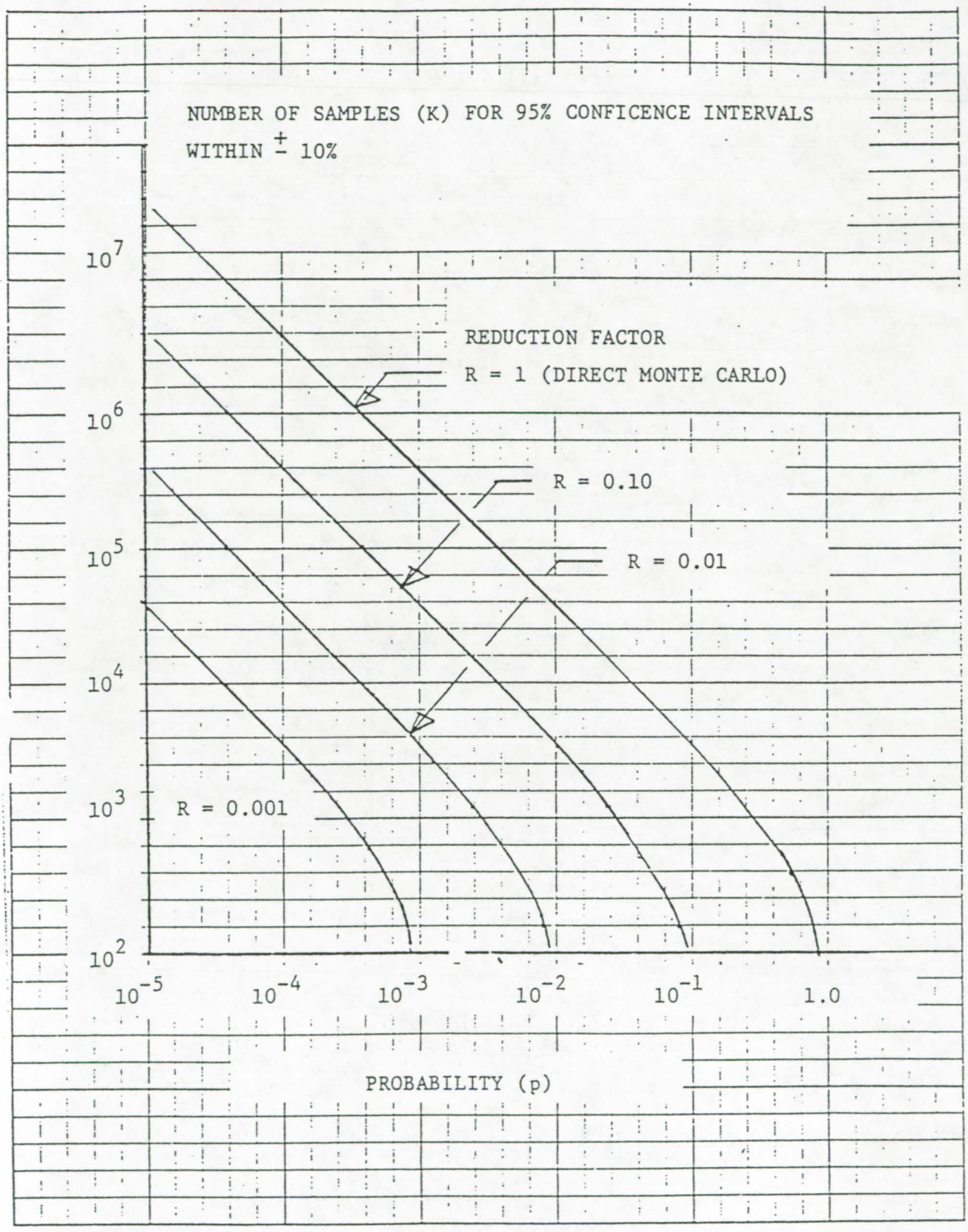


Fig. 5.6 Efficiency as a Function of Probability Level and Reduction Factor

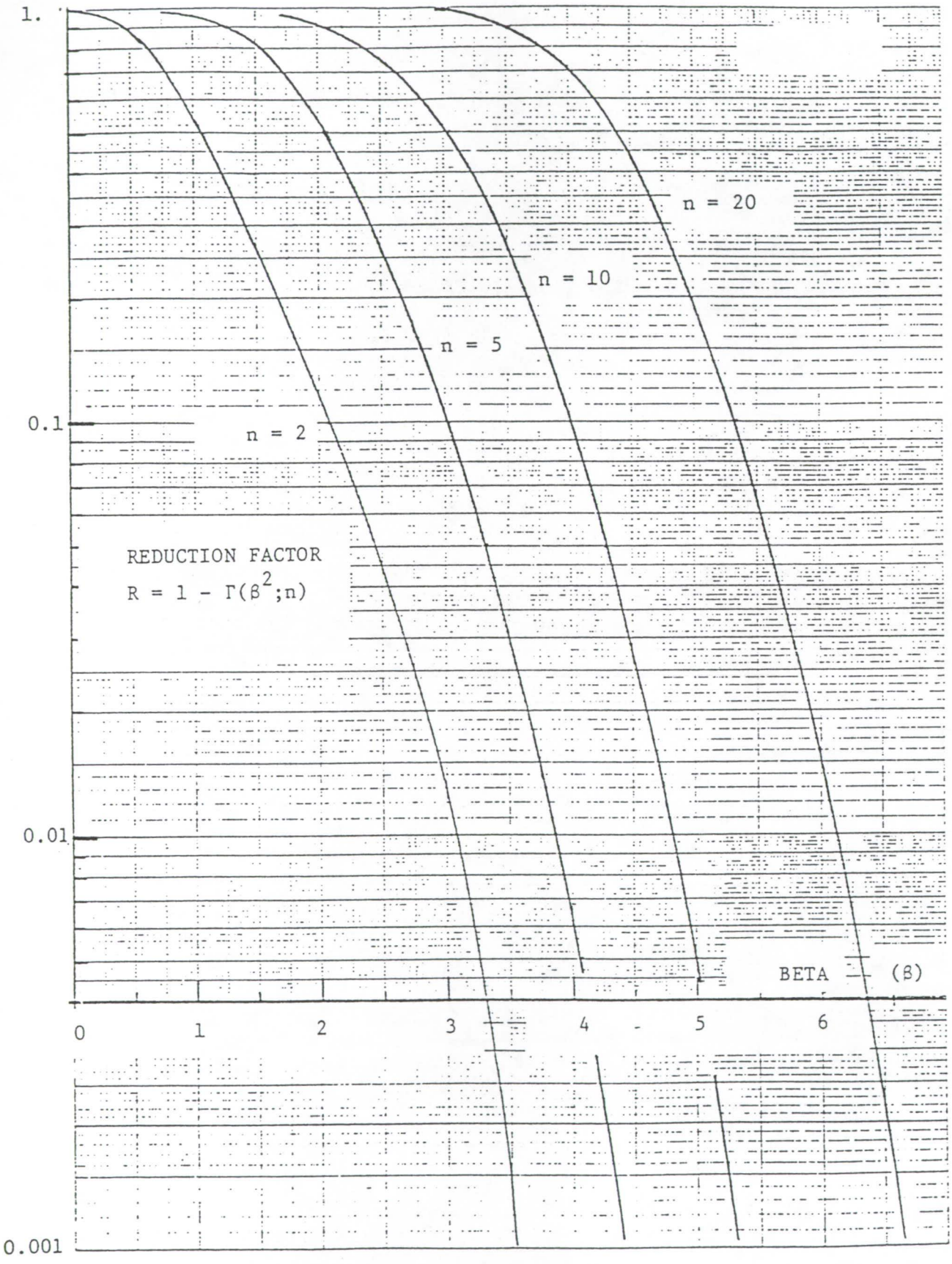


Fig. 5.7 Reduction Factor R as a Function of Beta and the Number of Variables.

and the scheme becomes more efficient. However, Fig. 5.6 does not tell the whole story with regard to Harbitz. Some extra computer time is required to run the Rachwitz-Fiessler analysis; and the sampling process may take more time as well.

As we now examine Fig. 5.7 in light of Fig. 5.6, we note that large R (poor efficiency) is associated with larger n and smaller β. In the latter case, as demonstrated in the example below, the loss of efficiency in small β (large R) is partially offset by higher probability levels (see Fig. 5.6).

In summary, the Harbitz method will always require a smaller sample relative to direct Monte Carlo. However, it is likely that direct Monte Carlo would require less running time for points with probability levels, say between 0.10 and 0.90. Location of these transition points are not known at this time, but they are not critical. But what is important is that Harbitz can be employed effectively for the tail regions of the distribution. The example illustrates why.

Example. An example which contrasts sample sizes by direct Monte Carlo with Harbitz is given in Table 5.1. The response function $Z(\underset{\sim}{X})$ is assumed to be a linear function of $\underset{\sim}{X}$. There are n = 5 variables and all X_i are assumed to be normal.

Required sample sizes K for estimates $\hat{p}_1$ which are within $\pm$ 10% of p with a confidence of 95% are shown in Table 5.1 for both the Harbitz and direct Monte Carlo methods. Clearly, Harbitz does much better at the lower levels of p.

But note how the number of factors n affect the requirement on K. At the point where $\beta = 4.4$, K must be 60,000 for n = 5. But for n = 10, $K \tilde{=} 10^6$, and for n = 20, $K \tilde{=} 2 \times 10^7$.

In summary, Harbitz becomes impractical for large n. This is an undesirable characteristic that it shares with all of the other Monte Carlo schemes.

EXAMPLE

TABLE 5.1 Number of Samples Required for Harbitz Compared to Conventional Monte Carlo for Different Probability Levels

Random Design Factors: n = 5

Response Function: Linear with normal variates

			K^*	
BETA (β)	R	p	Harbitz	Direct Monte Carlo
4.4	.001	5.5E-6	60,000	6×10^7
3.7	.01	1.0E-4	40,000	4×10^6
3.0	0.10	1.4E-3	25,000	200,000
2.0	0.50	2.3E-2	10,000	20,000

*This is the approximate minimum sample size required for an estimate $\hat{p}_1$ to be within $\pm$ 10% of p with a confidence of 95%.

6.0 SUMMARY: A COMPARISON OF THE PERFROMANCE OF MONTE CARLO METHODS FOR PROBABILITY ESTIMATES

6.1 The Methods Studied

Computer programs were developed for the following Monte Carlo methods:

1. Conventional Monte Carlo
2. Variance reduction
3. Mean value method with stratified sampling
4. The Harbitz method

Each program was verified using several example problems. The performance of each method was studied. Specifically, computer CPU time to produce a point probability estimate within $\pm$ 10% of the exact value with 95% confidence is measured.

6.2 A Summmary of the Performance of Each Method

Results of the performance study are summarized in Fig. 6.1 where CYBER 175 CPU time is plotted as a function of probability level β and number of variables, n. It is important to note that β is related to the tail probability level p by

$$p = \Phi(-\beta) \tag{6.1}$$

where Φ is the standard normal CDF. Computer time for each method depends on factors other than probability level and number of variables. The distribution type for each factor and the form of the response function influence computation time. Therefore, the curves of Fig. 6.1 must be interpreted as characterizing the relationships for purposes of comparison.

6.3 Commentary on the Implications of Fig. 6.1

Several general conclusions can be made regarding the results presented in Fig. 6.1.

1. Fast probability integration (e.g., the Wu/FPI method) is far more efficient than Monte Carlo.

2. Variance reduction does not appear to be competitive with the other methods.

3. For small numbers of variables, the mean value and Harbitz methods are very efficient with the Harbitz method having a slight edge.

4. Computing time for both the mean value and Harbitz methods increases sharply as the number of variables increase.

5. For small numbers of variables, conventional Monte Carlo is not efficient. But the increase in computing time increases linearly with the number of variables. Because these curves are flatter than the mean value or Harbitz curves, conventional Monte Carlo actually becomes more efficient relative to each of these methods above a given n.

6. Conventional Monte Carlo gets very expensive as the probability level decreases. Note that the $\beta = 4$ curve is off of the chart.

7. One feature of conventional Monte Carlo is that a full sample of the response variable is generated. Therefore, the entire CDF of the response variable can be generated. On the other hand, several probability points have to be computed using the other methods. And the accuracy will be better for larger probability levels and worse for smaller p.

In summary, a general conclusion is that the Harbitz method seems to be the preferred approach. Note however, as the probability level p gets larger (and β smaller), the Harbitz method approaches conventional Monte Carlo. This can be seen from Fig. 5.7 in which $R \to 1$ as $\beta \to 0$.

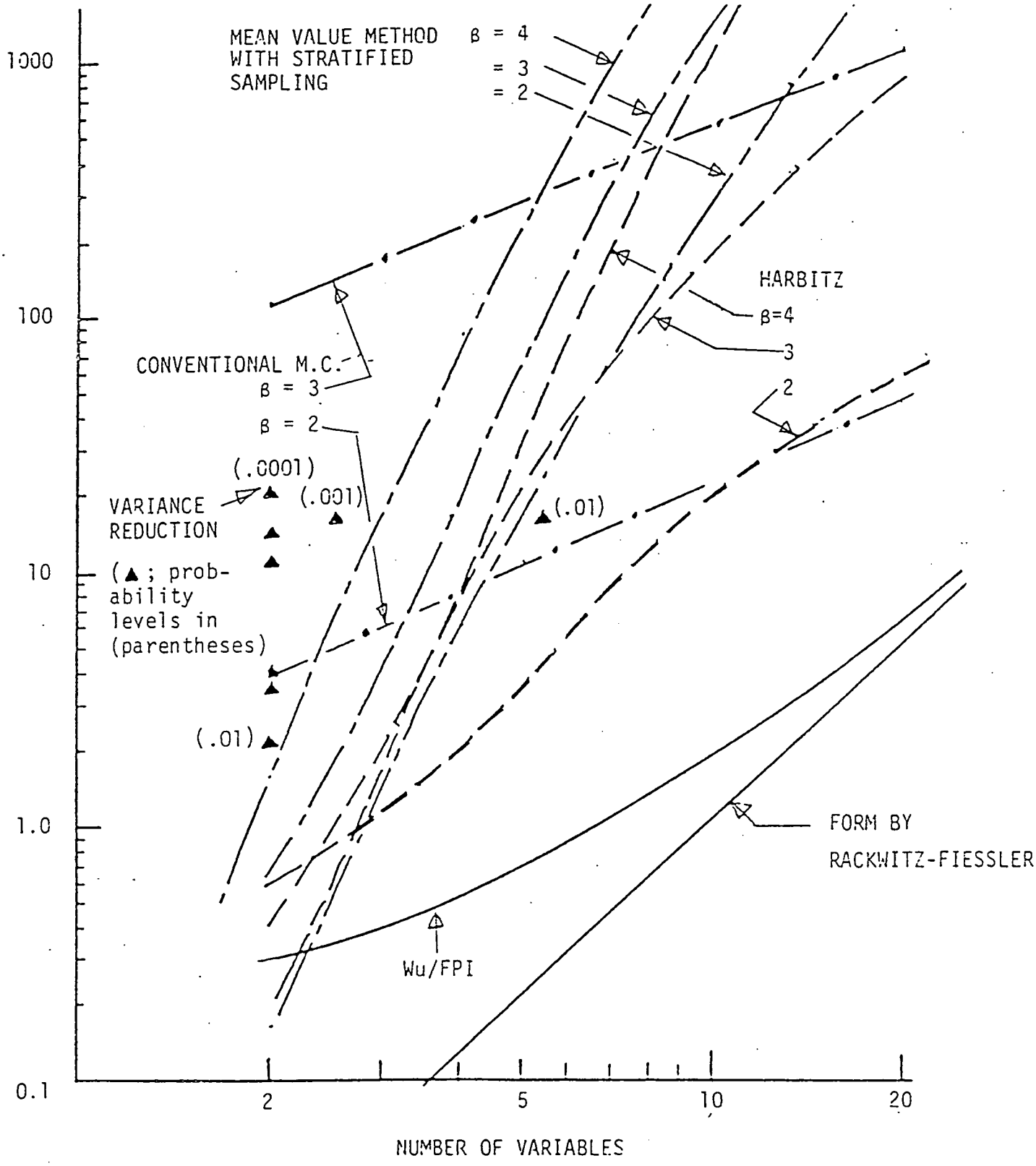


Fig. 6.1 A Summary of Efficiencies of Four Monte Carlo Methods for Computing Point Probabilities (for Monte Carlo $\pm$10% accuracy with a confidence of 95%).

References:

1. Ang, A. H. S. and Tang, W., Probability Concepts in Engineering Planning and Design, Vol. II, Wiley, 1984.

2. Box, G. E. P. and Muller, M. E., "A Note on the Generation of Random Normal Deviates," Annals of Math. Stat., 29, 1958, pp. 610-611.

3. Gumbel, E. J., Statistics of Extremes, Columbia U. Press, 1958

4. Hahn, G. J. and Shapiro, S. S., Statistical Models in Engineering, Wiley, 1968.

5. Hines, W. H. and Montgomery, D. C., Probability, Statistics, in Engineering and Management Science, 2nd Ed., Wiley, 1972.

6. Mann, N. R. et al., Methods for Statistical Analysis of Reliability and Life Data, Wiley, 1974.

7. Wirth, N., Algorithms + Data Structures = Programs, Prentice-Hall, 1976.

8. Harbitz, A., "An Efficient Sampling Method for Probability of Failure Calculation," Structural Safety, 3, (1986), 109-115.

APPENDIX A. RANDOM SAMPLES FROM GIVEN DISTRIBUTIONS

Following are the algorithms used to generate random variates from the normal, lognormal, Weibull, extreme value (Type I), and the Frechet distributions. The computer, using a congruential algorithm, samples random numbers u_i from a uniform distribution U(0,1). Forms given below transform uniform variates to variates X_i of other models.

Antithetic variates x_i' (defined as having a negative correlation to x_i) are generated as shown. These antithetic variates are used in the variance reduction method described in Section 3.0.

A. Normal distribution, $N(\mu, \sigma)$; sample two uniform variates, u_i and u_{i+1}. Use the Box-Muller algorithm [1 , 2].

$$x_i = \left[\sqrt{-2\ \ell n(u_i)}\ \cos(2\pi\ u_{i+1})\right]\ \sigma + \mu$$

$$x_i' = -x_i + 2\mu$$

B. Lognormal distribution, $LN(\tilde{X},\ C_X)$; sample two uniform variates, u_i and u_{i+1}. Use the Box-Muller algorithm [1 , 2].

$$\sigma_X = \sqrt{\ell n\ (1 + C_X^2)}$$

$$\mu_X = \ell n\ \tilde{X}$$

$$x_i = \left[\exp(\sqrt{-2\ \ell n(u_i)}\ \cos\ (2\pi\ u_{i+1})\right]\ \sigma_X + \mu_X$$

$$x_i' = \exp(-x_i + 2\ \mu_X)$$

C. Weibull distribution

$$F_X(x) = 1 - \exp\ (\ -\ (\frac{x}{\beta})^{\alpha}\ ) = u \sim U[0,1]$$

$$1 - u = \exp\ (\frac{x}{\beta})^{\alpha}\ ) \sim\ U[0,1]$$

$$-\ \ell n\ (1 - u) = (\frac{x}{\beta})^{\alpha}$$

Thus,

$$x_i = \beta(-\ \ell n\ (1 - u_i)^{1/\alpha}$$

$$x_i' = \beta(-\ \ell n\ (u_i))^{1/\alpha}$$

D. EVD distribution

$$F_X(x) = \exp(-\exp(-\alpha(X - \beta)) = u \sim U[0,1]$$

$$\exp(-\alpha(x - \beta)) = -\ln u$$

$$-\alpha(x - \beta) = \ln(-\ln u)$$

Thus,

$$x_i = \beta - \frac{1}{\alpha} \ln(-\ln(u_i))$$

$$x_i' = \beta - \frac{1}{\alpha} \ln(-\ln(1 - u_i))$$

E. Frechet distribution

$$F_X(x) = \exp -\left(\frac{v}{x}\right)^k = u \sim U[0,1]$$

$$\left(\frac{v}{x}\right)^k = -\ln(u)$$

Thus,

$$x_i = v\,(-\ln(u_i))^{-1/k}$$

$$x_i' = v\,(-\ln(1 - u_i))^{-1/k}$$

APPENDIX B. LISTING OF CONVENTIONAL MONTE CARLO PROGRAM (COMOC)

This version runs on the VAX and the CYBER 175. It is not interactive.

The performance function is introduced in subroutine LSFMC as XA.

See listing.

Card 1 Limit state function (not used in program; only printed on output)

Card 2 Number of trials; number of variables (free format)

Card 3 PLOT and ISTD type

PLOT: Y_i's are sorted to construct empirical CDF

0 = no sort

1 = Y_i's are sorted

ISTD; option to enter standard deviations or coefficients of deviations or coefficients of variation of each variable (if lognormal, always use COV).

0 = COV

1 = Std. dev.

Now enter each variable, its distribution type, and its moments.

Card 4 Variable name.

Card 5 Distribution, mean, and standard deviation

1 = WEI (Weibull)

2 = NORM (Normal

3 = EVD (Extreme value distribution)

4 = LN (Lognormal; always use median and COV)

5 = FRE (Frechet)

Then repeat 4 and 5 for all of the other variables.

```
      PROGRAM GMC(INPUT,OUTPUT,TAPE5=INPUT,TAPE6=OUTPUT)
        IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      DIMENSION INAME(20),XMEAN(20),XSTD(20),DIST(20),DTRANS(20),X(20)
      DIMENSION Y(10000),F(5),AL(20),BE(20)
        COMMON /TWO/ PI,PI2,SPI2
        CHARACTER*80 GRS,FIN,FOUT,AA*7,BB*6,CC*3,DD*3,EE*7
        CHARACTER*7 INAME,DTRANS
        DATA AA/'WEIBULL'/
        DATA BB/'NORMAL'/
        DATA CC/'EVD'/
        DATA DD/'LOG'/
        DATA EE/'FRECHET'/
C       CALL RANSET(0) FOR CYBER
        CALL RANSET(0)
8004    CONTINUE
        READ(5,'(A)',END=8888) GRS
        READ(5,*) K,N
        READ(5,*) PLOT,ISTD
C       READ(5,*) ISEED FOR VAX
        DO 7901 I=1,N
        READ(5,'(A)') INAME(I)
        READ(5,*) DIST(I),XMEAN(I),XSTD(I)
7901    CONTINUE
        IF(ISTD.EQ.0) THEN
        DO 913 I=1,N
        IF(DIST(I).EQ.4.) GO TO 913
        XSTD(I)=XMEAN(I)*XSTD(I)
913     CONTINUE
        END IF
      IF(K.GT.10000) K=10000
C
C
        DO 1234 I=1,N
        AL(I)=0.D0
        BE(I)=0.D0
1234    CONTINUE
        PI=4.D0*DATAN(1.D0)
        PI2=PI+PI
        SPI2=1.D0/DSQRT(PI2)
      DO 1 I=1,N
      IF(DIST(I).EQ.1.) DTRANS(I)=AA
      IF(DIST(I).EQ.2.) DTRANS(I)=BB
      IF(DIST(I).EQ.3.) DTRANS(I)=CC
      IF(DIST(I).EQ.4.) DTRANS(I)=DD
      IF(DIST(I).EQ.5.) DTRANS(I)=EE
        IF(DIST(I).EQ.1.) CALL WEI(XMEAN(I),XSTD(I),AL(I),BE(I))
        IF(DIST(I).EQ.3.) CALL EVD(XMEAN(I),XSTD(I),AL(I),BE(I),PI)
        IF(DIST(I).EQ.5.) CALL FRE(XMEAN(I),XSTD(I),AL(I),BE(I))
    1 CONTINUE
C
C*  THE DATA IS PRINTED OUT.
C
      WRITE(6,11) GRS,K,N
      WRITE(6,12)
      WRITE(6,13)(INAME(I),DTRANS(I),XMEAN(I),XSTD(I),I=1,N)
C       GENERATE RANDOM # AND CORRESPONDING RANDOM VARIABLE
        NUM=0
      DO 4 I=1,K
      DO 3 J=1,N
      CALL GENX(DIST(J),AL(J),BE(J),X(J),XMEAN(J),XSTD(J),ISEED)
```

CONVENTIONAL MONTE CARLO PROGRAM (COMOC): Runs on the VAX or CYBER 175

```
   CONTINUE
    CALL LSFMC(Y(I),N,X)
    IF(Y(I).LE.0.D0) NUM=NUM+1
4 CONTINUE

SORT Y

 CALL STAT(Y,K,YMEAN,YSTD,YMED,YCOV)
 WRITE(6,15)YMEAN,YSTD,YMED,YCOV
 ROUTINE TO ACCUMULATE NUMBER OF TRIALS WITH NEGATIVE Y(I)
 VALUES AND PRINT OUT RESULTS

  RATIO = DBLE(NUM)/DBLE(K)
  WRITE(6,9) NUM,RATIO
  FORMAT(/,10X,'NUMBER OF NEG Y VALUES=',I5,'.',4X,
+'PERCENT OF TRIALS=',F9.6)
  IF(PLOT.EQ.0.) GO TO 3456
  CALL QSORT(Y,K)

 THE SORTED VALUE OF Y AND THE EMPIRICAL CDF ARE PRINTED.

    WRITE(6,1017)
    FORMAT(////,14X,'SORTED VALUES OF Y AND THE EMPIRICAL CDF',/)
    J1=1
    J2=5
    WRITE(6,1003) J1,(Y(I),I=J1,J2)
    FORMAT(1X,'I = ',I5,5E13.5)
    J1=J1+5
    J2=J2+5
    IF(J1.GT.K) GO TO 3031
    IF(J2.GT.K) THEN
    J2=K
    GO TO 3030
    END IF
    GO TO 3030
    CONTINUE
    WRITE(6,67)
    FORMAT(/)
    J=0
    J1=1
    DO 1009 I=1,K
    J=J+1
    F(J)=(DBLE(I)-.5)/DBLE(K)
    IF(J.EQ.5.OR.I.EQ.K) THEN
    WRITE(6,1003) J1,(F(L),L=1,J)
    J=0
    J1=J1+5
    END IF
    CONTINUE
    CONTINUE

   )RMAT(5(/),30X,'MONTE CARLO SOLUTION',5(/),10X,
   LIMIT STATE FUNCTION : ',A,5(/),10X,
   SAMPLE SIZE, K=',I7//10X,'NUMBER OF RANDOM VARIABLES, N=',I3///)
   )RMAT(26X,'RANDOM VARIABLES',//10X,'VARIABLE',2X,
   )ISTRIBUTION',8X,'MEAN',12X,'STD DEV')
   IRMAT(/11X,A7,5X,A7,5X,E12.5,5X,E12.5)
   IRMAT(/////10X,'STATISTICS OF Y :'//10X,'MEAN    =',E13.5//10X,
   STD DEV =',E13.5//10X,'MEDIAN  =',E13.5//10X,'COV     =',
   3.5,4(/))
```

```
 17 FORMAT(1H1,2(/),14X,'SORTED VALUES OF Y AND THE EMPIRICAL CDF')
 19 FORMAT((5E13.5))
      GO TO 8004
888   CONTINUE
125 STOP
    END
    SUBROUTINE STAT(U,M,XM,STD,XMED,COV)

*  THIS SUBROUTINE CALCULATES THE STATISTICS (MEAN,STD DEV,MEDIAN,COV)
*  OF Y FUNCTION.

      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
    DIMENSION U(M)
    XK=M
    XM=0.
    DO 63 I=1,M
    XM=XM+U(I)
 63 CONTINUE
    XM=XM/XK
    STD=0.
    DO 64 I=1,M
    STD=STD+(U(I)-XM)**2
 64 CONTINUE
    STD=STD/(XK-1.DO)
    STD=DSQRT(STD)
    COV=STD/XM
    XMED=XM/DSQRT(1.DO+COV**2)
    RETURN
    END
      SUBROUTINE GENX(DIST,ALPHA,BETA,X,XMEAN,XSTD,ISEED)
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      COMMON /TWO/ PI,PI2,SPI2
C
      IDIST=INT(DIST+.1)
C     AA=RAN(ISEED) FOR VAX
      AA=RANF()
      GO TO (1,2,3,4,5), IDIST
1     X=BETA*(-DLOG(AA))**(1.DO/ALPHA)
      RETURN
C2    BB=RAN(ISEED) FOR VAX
2     BB=RANF()
      E=DSQRT(-2.DO*DLOG(AA))
      X=E*DCOS(PI2*BB)*XSTD+XMEAN
      RETURN
3     X=BETA-DLOG(-DLOG(AA))/ALPHA
      RETURN
C4    BB=RAN(ISEED) FOR VAX
4     BB=RANF()
      SDX=DSQRT(DLOG(1.DO+XSTD**2))
      UX=DLOG(XMEAN)
      E=DSQRT(-2.DO*DLOG(AA))
      X=DEXP(E*DCOS(PI2*BB)*SDX+UX)
      RETURN
5     X=BETA*(-DLOG(AA))**(-1.DO/ALPHA)
      RETURN
      END
      SUBROUTINE BISECT(COV,ISIGN,ALPHA)
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
C     ISIGN = 1; WEIBULL DIST.
C           = 2; FRECHET DIST.
```

GENX obtains random samples from distributions

RAN is library uniform random number generator for CYBER 175

```
F(X,COV)=-(1.DO+COV**2)*GAMMA(X)**2+GAMMA(2.*X)
IF(ISIGN.EQ.1) X1=COV**(1.08)
IF(ISIGN.EQ.2) X1=COV**(.677)/2.33
IF(ISIGN.EQ.2.AND.X1.GT..49DO) X1=.48999999
IF(ISIGN.EQ.1) F1=F(X1,COV)
IF(ISIGN.EQ.2) F1=F(-X1,COV)
IF(DABS(F1).LE.1.D-10) GO TO 1
X2=X1+.01DO
IF(ISIGN.EQ.1) F2=F(X2,COV)
IF(ISIGN.EQ.2) F2=F(-X2,COV)
F12=F1*F2
IF(F12.LT.0.) GO TO 20
IF(DABS(F1).GT.DABS(F2)) X1=X2
IF(DABS(F1).LT.DABS(F2)) X1=X1-.01DO
GO TO 7
CONTINUE
X3=(X1+X2)*.5DO
IF(ISIGN.EQ.1) F13=F(X1,COV)*F(X3,COV)
IF(ISIGN.EQ.2) F13=F(-X1,COV)*F(-X3,COV)
IF(F13.LT.0.) X2=X3
IF(F13.GT.0.) X1=X3
DX=DABS(X1-X2)
IF(DX.GE.1.D-9) GO TO 2
ALPHA=1.DO/X1
RETURN
END
SUBROUTINE WEI(XMEAN,XDEV,ALPHA,BETA)
IMPLICIT DOUBLE PRECISION (A-H,O-Z)
COV=XDEV/XMEAN
CALL BISECT(COV,1,ALPHA)
AL1=1.DO/ALPHA
BETA=XMEAN/GAMMA(AL1)
RETURN
END
SUBROUTINE FRE(XMEAN,XDEV,ALPHA,BETA)
IMPLICIT DOUBLE PRECISION (A-H,O-Z)
COV=XDEV/XMEAN
CALL BISECT(COV,2,ALPHA)
AL1=1.DO/ALPHA
BETA=XMEAN/GAMMA(-AL1)
RETURN
END
SUBROUTINE EVD(XMEAN,STD,ALPHA,BETA,PI)
IMPLICIT DOUBLE PRECISION (A-H,O-Z)
ALPHA=PI/(STD*DSQRT(6.DO))
BETA=XMEAN-.57721566490153/ALPHA
RETURN
END
DOUBLE PRECISION FUNCTION GAMMA(Y1)
IMPLICIT DOUBLE PRECISION (A-H,O-Z)
COMMON /TWO/ PI,PI2,SPI2
X=Y1+1.D+0
Z=X
IF(X.GE.6.0D+0)GO TO 456
N=INT(X)
Z=(6.0D+0)-N+X
Y=1.D+0/Z**2
ALG=(Z-.5D+0)*DLOG(Z)+.5D+0*DLOG(PI2)-
Z-(1.D+0/(12.D+0*Z))*(((Y/0.14D+3-1.D+0/0.105D+3)*Y+
1.D+0/.3D+2)*Y-1.D+0)
```

BISECT used to compute Weibull and Frechet shape parameter (exponent)

Computes Weibull parameters

Computes Frechet parameters

Computes EVD parameters

The gamma function

```
      IF(X.GE.6.D+0)GO TO 457
      ITE=6-N
      DO 3 J=1,ITE
      A=X+J-1.D+0
      ALG=ALG-DLOG(A)
 3    CONTINUE
457   GAMMA=DEXP(ALG)
      RETURN
      END
      SUBROUTINE QSORT(A,N)
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      DIMENSION A(N),KSL(240),KSR(240)
      KS=1
      KSL(1)=1
      KSR(1)=N
10    CONTINUE
      L=KSL(KS)
      KR=KSR(KS)
      KS=KS-1
20    CONTINUE
      I=L
      J=KR
      LR=(L+KR)/2
      X=A(LR)
30    CONTINUE
      IF(A(I).LT.X) THEN
      I=I+1
      GO TO 30
      END IF
40    CONTINUE
      IF(X.LT.A(J)) THEN
      J=J-1
      GO TO 40
      END IF
      IF(I.LE.J) THEN
      W=A(I)
      A(I)=A(J)
      A(J)=W
      I=I+1
      J=J-1
      END IF
      IF(I.LE.J) GO TO 30
      IF(I.LT.KR) THEN
      KS=KS+1
      KSL(KS)=I
      KSR(KS)=KR
      END IF
      KR=J
      IF(L.LT.KR) GO TO 20
      IF(KS.NE.0) GO TO 10
      RETURN
      END
      SUBROUTINE LSFMC(XA,N,X)
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      DIMENSION X(N)
      XA=X(1)-X(2)
      RETURN
      END
```

This is the sort routine, QUICKSORT

This is where the limit state is introduced

MONTE CARLO SOLUTION

LIMIT STATE FUNCTION : R=S

SAMPLE SIZE, K= 10000

NUMBER OF RANDOM VARIABLES, N= 2

RANDOM VARIABLES

VARIABLE	DISTRIBUTION	MEAN	STD DEV
R	NORMAL	.50000E+02	.50000E+01
S	NORMAL	.20000E+02	.12000E+02

STATISTICS OF Y :

MEAN = .30027E+02

STD DEV = .13060E+02

MEDIAN = .27535E+02

COV = .43493E+00

NUMBER OF NEG Y VALUES= 94. PERCENT OF TRIALS= .009400

```
:=S
10000,2
0.,0
R
2.,50.,.1
S
2.,20.,.6
```

APPENDIX C. THE SORT ROUTINE: "QUICKSORT"

QUICKSORT is described in detail in the book by Wirth [7], who describes its performance as "spectacular," and claims that it is the best sorting method on arrays known so far. The method is based on exchanges and the inventor C.A.R.Hoare recognized that sorting becomes most efficient when exchanges are made over large distances.

The table below shows execution times (in milliseconds) consumed by several proposed sorting methods as executed by the PASCAL system on a CDC 6400 computer. The three columns contain times used to sort the already ordered array, a random permutation, and the inversely ordered array. The left figure in each column is for 256 items, and right one for 512 items.

In summary, the computational effort needed for QUICKSORT is of the order of n log n.

	Ordered		Random		Inversely Ordered	
Straight insertion	12	23	366	1444	704	2836
Binary insertion	56	125	373	1327	662	2490
Straight selection	489	1907	509	1956	695	2675
Bubblesort	540	2165	1026	4054	1492	5931
Bubblesort with flag	5	8	1104	4270	1645	6542
Shakersort	5	9	961	3642	1619	6520
Shellsort	58	116	127	349	157	492
Heapsort	116	253	110	241	104	226
Quicksort	31	69	60	146	37	79
Mergesort	99	234	102	242	99	232

Execution Times of Sort Programs.

APPENDIX D. LISTING OF THE VARIANCE REDUCTION MONTE CARLO PROGRAM (VARED)

This version runs on the VAX and the CYBER 175. It is not interactive.

The performance function is introduced in subroutine LSFMC, then compiled and linked to the rest of the program.

Data Input for the VAX Version Variance Reduction Program

Card 1 Limit State Function (not used for calculations in the program)

Ex: g = R - S or R = S, etc.

Card 2 Number of Trials (the preliminary value of K); Number of Variables; Maximum Error in Secant Method for Solution of Maximum Impact Variable (a small number)

Ex: 1000, 3, 1.D-6

or 10000,5,1.D-7

Card 3 Confidence Interval; Gamma; ISTD;

a. C.I. = 0 to 100 in percent: Ex: 90; implies 90% C.I.

b. Gamma $0 \leq \gamma \leq 1$, but typically choose γ from 0.05 to 0.20. See Eq. 3.21 ff.

c. ISTD = OPTION to enter standard derivations and coefficients of variation of each variable (for LN Dist, always use COV)

0 = COV

1 = Std. dev.

Card 4 Enter ISEED

Any integer number between 0 and 262,139 to start the random sampling.

Ex: 23, 579, etc.

Card 5 Enter variable name. (Free format)

Card 6 Enter corresponding distribution, mean, and standard deviation (if LN always input median and COV); Ex: 1, 20, 2

a. dist. = 1 = Weibull

2 = Normal

3 = EVD

4 = Lognormal (LN)

5 = Frechet

Then repeat 5 and 6 for all of the other variables.

```
PROGRAM GMC
  PROGRAM GMC(INPUT,OUTPUT,TAPE5=INPUT,TAPE6=OUTPUT) FOR CYBER
  IMPLICIT DOUBLE PRECISION (A-H,O-Z)
DIMENSION INAME(20),XMEAN(20),XSTD(20),DIST(20),DTRANS(20),X(20)
DIMENSION Y(10000),F(5),AL(20),BE(20),XA(20),TX(20),TS(20)
  COMMON /TWO/PI,SPI2,PI2
  CHARACTER*70 GRS,FIN,FOUT,AA*7,BB*6,CC*3,DD*3,EE*7
  CHARACTER*7 INAME,DTRANS
  DATA AA/'WEIBULL'/
  DATA BB/'NORMAL'/
  DATA CC/'EVD'/
  DATA DD/'LOG'/
  DATA EE/'FRECHET'/

  READ(5,'(A)',END=8888) GRS
  READ(5,*) K,N,EPS
  READ(5,*) ZAL,GAM,ISTD,PLOT
  FOR CYBER, CALL RANSET(0) AND SKIP ISEED
  READ(5,*) ISEED
  DO 7901 I=1,N
  READ(5,'(A)') INAME(I)
  READ(5,*) DIST(I),XMEAN(I),XSTD(I)
  CONTINUE
  CONTINUE
  IF(ISTD.EQ.0) THEN
  DO 913 I=1,N
  IF(DIST(I).EQ.4.) GO TO 913
  XSTD(I)=XMEAN(I)*XSTD(I)
  CONTINUE
  END IF
IF(K.GT.10000) K=10000


  DO 1234 I=1,N
  AL(I)=0.DO
  BE(I)=0.DO
  IF(DIST(I).EQ.4.) THEN
  TX(I)=XMEAN(I)*DSQRT(1.DO+XSTD(I)**2)
  TS(I)=TX(I)*XSTD(I)
  ELSE
  TX(I)=XMEAN(I)
  TS(I)=XSTD(I)
  END IF
  CONTINUE
  PI=4.DO*DATAN(1.DO)
  PI2=PI+PI
  SPI2=1.DO/DSQRT(PI2)
DO 1 I=1,N
IF(DIST(I).EQ.1.) DTRANS(I)=AA
IF(DIST(I).EQ.2.) DTRANS(I)=BB
IF(DIST(I).EQ.3.) DTRANS(I)=CC
IF(DIST(I).EQ.4.) DTRANS(I)=DD
IF(DIST(I).EQ.5.) DTRANS(I)=EE
  IF(DIST(I).EQ.1.) CALL WEI(XMEAN(I),XSTD(I),AL(I),BE(I))
  IF(DIST(I).EQ.3.) CALL EVD(XMEAN(I),XSTD(I),AL(I),BE(I),PI)
  IF(DIST(I).EQ.5.) CALL FRE(XMEAN(I),XSTD(I),AL(I),BE(I))
CONTINUE

: DATA IS PRINTED OUT.
```

Program VARED. Monte Carlo using variance reduction method; runs on the VAX or CYBER 175.

```
C       MAIN LOOP USING ANTITHETIC VARIANCE REDUCTION METHOD
C       FIND MAX. IMPACT VARIABLE
        DG=0.D0
        CALL LSFMC(G,N,TX)
        DO 700 I=1,N
        TX(I)=TX(I)+TS(I)
        CALL LSFMC(DGB,N,TX)
        DGA=DGB-G
        IF(DABS(DGA).LE.DABS(DG)) GO TO 701
        IV=I
        DG=DGA
701     TX(I)=TX(I)-TS(I)
700     CONTINUE
C
      WRITE(6,11) GRS,K,N
        WRITE(6,96) ZAL,GAM
96      FORMAT(10X,'CONFIDENCE INTERVAL = ',F6.2,' %',//,
     $  10X,'GAMMA = ',F6.2,///)
        WRITE(6,559) IV
559     FORMAT(10X,'MAX. IMPACT VARIABLE = X(',I2,')',/)
        IF(DG.LE.0.D0) WRITE(6,561)
561     FORMAT(10X,'VARIABLE TYPE IS STRESS',///)
        IF(DG.GT.0.D0) WRITE(6,563)
563     FORMAT(10X,'VARIABLE TYPE IS STRENGTH',///)
      WRITE(6,12)
      WRITE(6,13)(INAME(I),DTRANS(I),XMEAN(I),XSTD(I),I=1,N)
C       CALCULATE PROB. OF FAILURE
        K1=1
        K2=K
        ICO=1
98      CONTINUE
        DO 702 I=K1,K2
        DO 703 J=1,N
        IF(J.EQ.IV) GO TO 703
        CALL GENX(DIST(J),AL(J),BE(J),X(J),XA(J),XMEAN(J),XSTD(J),ISEED)
703     CONTINUE
        IF(DG.GT.0.D0) A=TX(IV)-3.D0*TS(IV)
        IF(DG.LE.0.D0) A=TX(IV)+2.D0*TS(IV)
        B=A+TS(IV)
        CALL SECA(EPS,A,B,IV,N,X)
        CALL CDFPDF(DIST(IV),AL(IV),BE(IV),X(IV),XMEAN(IV),XSTD(IV),
     $  1,CDF1,PDF)
        IF(DG.LE.0.D0) CDF1=1.D0-CDF1
        IF(DG.GT.0.D0) A=TX(IV)-3.D0*TS(IV)
        IF(DG.LE.0.D0) A=TX(IV)+2.D0*TS(IV)
        B=A+TS(IV)
        CALL SECA(EPS,A,B,IV,N,XA)
        CALL CDFPDF(DIST(IV),AL(IV),BE(IV),XA(IV),XMEAN(IV),XSTD(IV),
     $  1,CDF2,PDF)
        IF(DG.LE.0.D0) CDF2=1.D0-CDF2
        Y(I)=(CDF1+CDF2)*.5D0
702     CONTINUE
C
  123 CALL STAT(Y,K1,K2,YMEAN,YSTD,YMED,YCOV)
        IF(ICO.EQ.1) THEN
        YM=YMEAN
        YS=YSTD
        YME=YMED
        YC=YCOV
        YM1=YM
```

```
      ELSE
      YM=(K*YM1+(K2-K)*YMEAN)/K2
      YS1=YS**2*(K-1)+K*YM1**2+YSTD**2*(K2-K-1)+(K2-K)*YMEAN**2
      YS2=YS1-K2*YM**2
      YS=DSQRT(YS2/(K2-1))
      YC=YS/YM
      YME=YM/DSQRT(1.D0+YC**2)
      END IF
      ZAL1=.005D0*(100.D0+ZAL)
      ZAX=XINV(ZAL1)
      ZX=ZAX*YC/DSQRT(DBLE(K2))
      PL=YM*(1.D0-ZX)
      PU=YM*(1.D0+ZX)
      WRITE(6,176) YM,ZAL,PL,PU
      FORMAT(///,10X,'ESTIMATE OF P = ',E13.5,//,
     $  10X,F5.2,' % CONFIDENCE INTERVALS ARE',//,
     $  10X,'PL = ',E13.5,5X,'PU = ',E13.5,///)

      WRITE(6,15) YMEAN,YSTD,YMED,YCOV
    IF(PLOT.EQ.0.) GO TO 3456
      J1=1
      J2=5
      WRITE(6,1003) J1,(Y(I),I=J1,J2)
      FORMAT(1X,'I = ',I5,5E13.5)
      J1=J1+5
      J2=J2+5
      IF(J1.GT.K2) GO TO 3031
      IF(J2.GT.K2) THEN
      J2=K2
      GO TO 3030
      END IF
      GO TO 3030
      CONTINUE
      WRITE(6,67)
      FORMAT(/)
      J=0
      J1=1
      DO 1009 I=1,K2
      J=J+1
      F(J)=(DBLE(I)-.5)/DBLE(K2)
      IF(J.EQ.5.OR.I.EQ.K2) THEN
      WRITE(6,1003) J1,(F(L),L=1,J)
      J=0
      J1=J1+5
      END IF
      CONTINUE
      CONTINUE
      K1=K+1
      K2=(YC*ZAX/GAM)**2+1
      IF(ICO.EQ.1) WRITE(6,99) GAM,K2
      FORMAT(//,10X,'K FOR GAMMA = ',F6.2,' IS ',I6)
      ICO=ICO+1
      IF(ICO.EQ.2.AND.K2.GT.K) GO TO 98
    FORMAT(1H1,5(/),30X,'MONTE CARLO SOLUTION',5(/),10X,
    'LIMIT STATE FUNCTION : ',A,5(/),10X,
    'SAMPLE SIZE =',I7//10X,'NUMBER OF RANDOM VARIABLES =',I3//)
    FORMAT(26X,'RANDOM VARIABLES',//10X,'VARIABLE',2X,
    'DISTRIBUTION',8X,'MEAN',12X,'STD DEV')
    FORMAT(/11X,A7,5X,A7,5X,E12.5,5X,E12.5)
    FORMAT(/////10X,'STATISTICS OF P :'//10X,'MEAN    =',E13.5//10X,
```

```
     +'STD DEV =',E13.5//10X,'MEDIAN  =',E13.5,//10X,'COV     =',
     +E13.5,////)
        IF(ANS1.EQ.'F'.OR.ANS1.EQ.'f') GO TO 8300
        WRITE(6,8301)
8301    FORMAT(' DO YOU HAVE ANOTHER DATA SET ?(Y/N) ',$)
        READ(5,8001) ANS3
        IF(ANS3.EQ.'Y'.OR.ANS3.EQ.'y') GO TO 8304
8888    CONTINUE
  125 STOP
      END
        SUBROUTINE SECA(EPS,A,B,IV,N,X)
        IMPLICIT DOUBLE PRECISION (A-H,O-Z)
        DIMENSION X(N)
        X(IV)=A
        CALL LSFMC(U,N,X)
        X(IV)=B
        CALL LSFMC(V,N,X)
1       CONTINUE
        IF(DABS(X(IV)-A).GE.EPS) THEN
        X(IV)=B-V*(B-A)/(V-U)
        A=B
        B=X(IV)
        U=V
        CALL LSFMC(V,N,X)
        GO TO 1
        END IF
        RETURN
        END
        SUBROUTINE CDFPDF(DIST,ALPHA,BETA,X,XMEAN,XDEV,ICDF,CDF,PDF)
        IMPLICIT DOUBLE PRECISION (A-H,O-Z)
        COMMON /TWO/PI,SPI2,PI2
        IDIST=INT(DIST+.1)
      GO TO (1,2,3,4,5),IDIST
    1 RB=X/BETA
      EW=RB**ALPHA
      IF(EW.GT.200.) EW=200.
      EXPWEI=DEXP(-EW)
      CDF=1.DO-EXPWEI
      IF(ICDF.EQ.1) GO TO 10
      PDF=(ALPHA/BETA)*(EW/RB)*EXPWEI
      GO TO 10
    2 Z=(X-XMEAN)/XDEV
      CDF=CDFNOR(Z)
      IF(ICDF.EQ.1) GO TO 10
      PDF=SPI2*DEXP(-Z**2*.5DO)/XDEV
      GO TO 10
    3 EE=ALPHA*(X-BETA)
      IF(EE.GT.200.) EE=200.
      YY=DEXP(-EE)
      IF(YY.GT.200.) YY=200.
      CDF=DEXP(-YY)
      IF(ICDF.EQ.1) GO TO 10
      EY=EE+YY
      IF(EY.GT.200.) EY=200.
      PDF=ALPHA*DEXP(-EY)
      GO TO 10
    4 CX21=XDEV**2+1.DO
      YMEAN=DLOG(XMEAN)
      YDEV=DSQRT(DLOG(CX21))
      Z=(DLOG(X)-YMEAN)/YDEV
```

This defined the performance functio

This subroutine determines t
point at which the CDF is
evaluated for the maximum
impact variable

Evaluates
the CDF

```
  CDF=CDFNOR(Z)
  IF(ICDF.EQ.1) GO TO 10
  EZ=-(Z**2)*.5D0
  IF(EZ.LE.-200.) EZ=-200.
  PDF=SPI2*DEXP(EZ)/(YDEV*X)
  GO TO 10
    TEMP=(BETA/X)**ALPHA
    CDF=DEXP(-TEMP)
    IF(ICDF.EQ.1) GO TO 10
    PDF=CDF*TEMP*ALPHA/X
) RETURN
  END
  DOUBLE PRECISION FUNCTION XINV (Z)
    IMPLICIT DOUBLE PRECISION (A-H,O-Z)
    F(X,P1)=P1-CDFNOR(X)
  Y=Z
  IF(Z.GT.0.5D0) Y=1.D0-Z
  C0=2.515517D0
  C1=0.802853D0
  C2=0.010328D0
  D1=1.432788D0
  D2=0.189269D0
  D3=0.001308D0
    T=(-2.D0*DLOG(Y))**.5D0
  DNUM=C0+T*(C1+T*C2)
  DNOM=1.0D0+T*(D1+T*(D2+T*D3))
  X=T-(DNUM/DNOM)
  IF(Z.LT.0.5D0) X=-X
    A=X
    B=X+.001D0
    V=F(B,Z)
    U=F(A,Z)
    XX=B
    CONTINUE
    IF(DABS(XX-A).GE.1.D-10) THEN
    XX=B-V*(B-A)/(V-U)
    A=B
    B=XX
    U=V
    V=F(XX,Z)
    GO TO 1
    END IF
    XINV=XX
    RETURN
    END
    DOUBLE PRECISION FUNCTION CDFNOR(Z)
  ; FUNCTION COMPUTES THE NORMAL CDF.
    IMPLICIT DOUBLE PRECISION (A-H,O-Z)
    COMMON /TWO/PI,SPI2,PI2
  DATA A/0.31938153D0/,B/-0.356563782D0/,C/1.781477937D0/,
  D/-1.821255978D0/,E/1.330274429D0/
  EZ=-(Z**2)*.5D0
  CDFNOR=0.0D0
  IF(EZ.LE.-200.0D0) GO TO 1
  ZX=SPI2*DEXP(EZ)
    IF(DABS(Z).GT.6.D0) GO TO 2
  T=1.D0/(1.D0+(0.2316419D0*DABS(Z)))
  CDFNOR=ZX*T*(A+T*(B+T*(C+T*(D+T*E))))
    GO TO 1
    Z2=1.D0/(Z*Z)
```

The inverse normal using the secant method

```
      CDFNOR=ZX*(1.DO-Z2*(1.DO-3.DO*Z2*(1.DO-5.DO*Z2)))/DABS(Z)
    1 IF(Z.GT.0.0DO) CDFNOR=1.0DO-CDFNOR
      RETURN
      END
      SUBROUTINE STAT(U,K1,K2,XM,STD,XMED,COV)
C
C*  THIS SUBROUTINE CALCULATES THE STATISTICS (MEAN,STD DEV,MEDIAN,COV)
C*  OF Y FUNCTION.
C
        IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      DIMENSION U(K2)
      XK=K2-K1+1
      XM=0.
      DO 63 I=K1,K2
      XM=XM+U(I)
   63 CONTINUE
      XM=XM/XK
      STD=0.
      DO 64 I=K1,K2
      STD=STD+(U(I)-XM)**2
   64 CONTINUE
      STD=STD/(XK-1.DO)
        STD=DSQRT(STD)
        COV=STD/XM
        XMED=XM/DSQRT(1.DO+COV**2)
      RETURN
      END
        SUBROUTINE GENX(DIST,ALPHA,BETA,X,XA,XMEAN,XSTD,ISEED)
        IMPLICIT DOUBLE PRECISION (A-H,O-Z)
        COMMON /TWO/PI,SPI2,PI2
C
        IDIST=INT(DIST+.1)
C       FOR CYBER, AA=RANF()
        AA=RAN(ISEED)
        GO TO (1,2,3,4,5), IDIST
1       X=BETA*(-DLOG(AA))**(1.DO/ALPHA)
        XA=BETA*(-DLOG(1.DO-AA))**(1.DO/ALPHA)
        RETURN
2       BB=RAN(ISEED)
C       FOR CYBER, BB=RANF()
        E=DSQRT(-2.DO*DLOG(AA))
        X=E*DCOS(PI2*BB)*XSTD+XMEAN
        XA=-X+2.DO*XMEAN
        RETURN
3       X=BETA-DLOG(-DLOG(AA))/ALPHA
        XA=BETA-DLOG(-DLOG(1.DO-AA))/ALPHA
        RETURN
4       BB=RAN(ISEED)
C       FOR CYBER, BB=RANF()
        SDX=DSQRT(DLOG(1.DO+XSTD**2))
        UX=DLOG(XMEAN)
        W=DSQRT(-2.DO*DLOG(AA))*DCOS(PI2*BB)*SDX+UX
        X=DEXP(W)
        XA=DEXP(-W+2.DO*UX)
        RETURN
5       X=BETA*(-DLOG(AA))**(-1.DO/ALPHA)
        XA=BETA*(-DLOG(1.DO-AA))**(-1.DO/ALPHA)
        RETURN
        END
        SUBROUTINE SECA1(COV,ISIGN,ALPHA)
```

GENX obtains random samples from the distributions

RAN is library uniform random number generator for CYBER 175

```
IMPLICIT DOUBLE PRECISION (A-H,O-Z)
ISIGN = 1; WEIBULL DIST.
      = 2; FRECHET DIST.
F(X,COV)=-(1.DO+COV**2)*GAMMA(X)**2+GAMMA(2.*X)
IF(ISIGN.EQ.1) X1=COV**(1.08)
IF(ISIGN.EQ.2) X1=COV**(.677)/2.33
IF(ISIGN.EQ.2.AND.X1.GT..49DO) X1=.48999999
IF(ISIGN.EQ.1) F1=F(X1,COV)
IF(ISIGN.EQ.2) F1=F(-X1,COV)
IF(DABS(F1).LE.1.D-10) GO TO 1
X2=X1+.01DO
IF(ISIGN.EQ.1) F2=F(X2,COV)
IF(ISIGN.EQ.2) F2=F(-X2,COV)
XX=X2
CONTINUE
IF(DABS(XX-X1).GE.1.D-9) THEN
XX=X2-F2*(X2-X1)/(F2-F1)
X1=X2
X2=XX
F1=F2
IF(ISIGN.EQ.1) F2=F(XX,COV)
IF(ISIGN.EQ.2) F2=F(-XX,COV)
GO TO 10
END IF
X1=XX
ALPHA=1.DO/X1
RETURN
END
```

Secant method for computing Weibull and Frechet exponents

```
SUBROUTINE WEI(XMEAN,XDEV,ALPHA,BETA)
IMPLICIT DOUBLE PRECISION (A-H,O-Z)
COV=XDEV/XMEAN
CALL SECA1(COV,1,ALPHA)
AL1=1.DO/ALPHA
BETA=XMEAN/GAMMA(AL1)
RETURN
END
```

Computes Weibull parameters

```
SUBROUTINE FRE(XMEAN,XDEV,ALPHA,BETA)
IMPLICIT DOUBLE PRECISION (A-H,O-Z)
COV=XDEV/XMEAN
CALL SECA1(COV,2,ALPHA)
AL1=1.DO/ALPHA
BETA=XMEAN/GAMMA(-AL1)
RETURN
END
```

Computes Frechet parameters

```
SUBROUTINE EVD(XMEAN,STD,ALPHA,BETA,PI)
IMPLICIT DOUBLE PRECISION (A-H,O-Z)
ALPHA=PI/(STD*DSQRT(6.DO))
BETA=XMEAN-.57721566490153/ALPHA
RETURN
END
```

Computes EVD parameters

```
DOUBLE PRECISION FUNCTION GAMMA(Y1)
IMPLICIT DOUBLE PRECISION (A-H,O-Z)
COMMON /TWO/PI,SPI2,PI2
X=Y1+1.D+0
Z=X
IF(X.GE.6.0D+0)GO TO 456
N=INT(X)
Z=(6.0D+0)-N+X
Y=1.D+0/Z**2
ALG=(Z-.5D+0)*DLOG(Z)+.5D+0*DLOG(PI2)-
```

The gamma function

```
     $  Z-(1.D+0/(12.D+0*Z))*(((Y/0.14D+3-1.D+0/0.105D+3)*Y+
     $  1.D+0/.3D+2)*Y-1.D+0)
        IF(X.GE.6.D+0)GO TO 457
        ITE=6-N
        DO 3 J=1,ITE
        A=X+J-1.D+0
        ALG=ALG-DLOG(A)
   3    CONTINUE
 457    GAMMA=DEXP(ALG)
        RETURN
        END
```

Note: The performance function must be introduced in subroutine LSFMC. For an example of subroutine LSFMC, see the last page of Appendix B.

Appendix E

Program SELSAM: The Mean Value Method for Evaluating a Multiple Integral, and Enhancement by Stratified Sampling

How the Data is Input . . .

1. NK, NT, N

   NK = 0; Hasofer-Lind design point for reference

   NK = 1; Rackwitz-Fiessler design point for reference

   NT = Total number of samples

   N = Total number of random variables

2. ISTRIP

   ISTRIP = Total number of STRIPS (or boxes) for stratified sampling;

   ISTRIP = 1 gives you the "mean value" method

3. BOX (I), IBOX (I) This is repeated for each box.

   BOX (I) = ith strip length from the reference point (design point) in standard deviations.

   IBOX (I) = trial points in ith strip

4. VAR (I); this along with the next line will be repeated for each random variable

   VAR (I) = ith random variable name

5. IDIST (I) XMEAN (I), STD (I); this corresponds to VAR (I)

   IDIST (I) = ith random variable distribution

   1 = WEIBULL

   2 = NORMAL

   3 = EVD

   4 = LOGNORMAL

   5 = FRECHET

XMEAN (I) = ith random variable mean value

STD (2) = ith random variable standard deviation

*If nognormal, median and COV instead of mean and std.

6. 20 = constant in performance function; this allows the user to make an easy change in the performance function when constructing a CDF

7. The user should supply the LSFRA and G function in the last section of program (see example)

```
PROGRAM SELSAM(INPUT,OUTPUT,TAPE5=INPUT,TAPE6=OUTPUT)
IMPLICIT DOUBLE PRECISION (A-H,O-Z)
DIMENSION XMEAN(20),STD(20),IDIST(20),XR(20),AL(20),BE(20)
DIMENSION X(20),BOX(20),IBOX(20),IDIV(20),AREA(20),ITEST(20)
DIMENSION IDIV1(20),TMEAN(20),TSTD(20),Z(20),UX(20),TCOV(20)
CHARACTER*7 VAR(20),AA
COMMON /TWO/ PI,PI2,SPI2
COMMON /RAC/ NK
CALL RANSET(0)
PI=4.DO*DATAN(1.DO)
PI2=PI+PI
SPI2=1.DO/DSQRT(PI2)

EPSI = STOP CRITERION IN RACA

EPSI=1.D-4

NK = 0; H-L
NK = 1; R-F
NT; NUMBER OF TRIALS
N; NUMBER OF RANDOM VARIABLES

READ(5,*) NK,NT,N

ISTRIP; NUMBER OF STRIPS

READ(5,*) ISTRIP

DO 300 I=1,ISTRIP

BOX; DISTANCE FROM ORIGIN FOR i-TH STRIP (MULTIPLIED BY SIGMA)
IBOX; NUMBER OF POINTS IN i-TH BOX

READ(5,*) BOX(I),IBOX(I)

CONTINUE

CALCULATE EACH STRIP AREA

DO 150 I=1,ISTRIP-1
AREA(I)=(2.DO*BOX(I))**N-(2.DO*BOX(I+1))**N
CONTINUE
AREA(ISTRIP)=(2.DO*BOX(ISTRIP))**N
WRITE(6,769)
DO 1 I=1,N
AL(I)=0.DO
BE(I)=0.DO

ENTER VARIABLE NAME

READ(5,'(A)') VAR(I)

ENTER DISTRIBUTION, MEAN, AND STANDARD DEVIATION
IF LN, USE MEDIAN AND COV

READ(5,*) IDIST(I),XMEAN(I),STD(I)

IF(IDIST(I).EQ.1) AA='WEIBULL'
IF(IDIST(I).EQ.2) AA='NORMAL'
```

```
      IF(IDIST(I).EQ.3) AA='EVD'
      IF(IDIST(I).EQ.4) AA='LOG N.'
      IF(IDIST(I).EQ.5) AA='FRECHET'
      GO TO (101,17,103,17,105) ,IDIST(I)
101   CALL WEI(XMEAN(I),STD(I),AL(I),BE(I))
      GO TO 17
103   CALL EVD(XMEAN(I),STD(I),AL(I),BE(I),PI)
      GO TO 17
105   CALL FRE(XMEAN(I),STD(I),AL(I),BE(I))
17    CONTINUE
      WRITE(6,768) VAR(I),AA,XMEAN(I),STD(I)
1     CONTINUE
      WRITE(6,767)
      WRITE(6,766) NT
      WRITE(6,7661) (IBOX(I),I=1,ISTRIP)
      WRITE(6,7662) (BOX(I),I=1,ISTRIP)
      DO 7 I=1,N
         IF(IDIST(I).EQ.4) THEN
            TMEAN(I)=XMEAN(I)*DSQRT(1.DO+STD(I)**2)
            TSTD(I)=TMEAN(I)*STD(I)
            TCOV(I)=STD(I)
         ELSE
            TMEAN(I)=XMEAN(I)
            TSTD(I)=STD(I)
            TCOV(I)=TMEAN(I)/TSTD(I)
         END IF
7     CONTINUE
            DO 665 K1=1,N
            IF(IDIST(K1).EQ.4) THEN
            TMEAN(K1)=DLOG(TMEAN(K1))
            TSTD(K1)=DSQRT(DLOG(1.DO+TCOV(K1)**2))
            END IF
665   CONTINUE

C     ZO = CONSTANT IN LSFRA

            READ(5,*) ZO

            CALL RACA(Z,N,XR,EPSI,TMEAN,IDIST,TSTD,TCOV,AL,BE,BET,ZO)
            WRITE(6,61)
            WRITE(6,63) (XR(KKJ),KKJ=1,N)
            DO 666 K1=1,N
            IF(IDIST(K1).EQ.4) THEN
            TMEAN(K1)=DEXP(TMEAN(K1)+TSTD(K1)**2*.5)
            TSTD(K1)=TMEAN(K1)*TCOV(K1)
            END IF
666   CONTINUE
333   SUMP=0.DO
      SUMS=0.DO
      WRITE(6,770)
      DO 100 IJ=1,NT
      CALL SECOND(TX1)
      SUM=0.DO
      IDIV1(1)=IBOX(1)
      DO 15 L=1,ISTRIP
      IDIV(L)=0
      SUMX=0.DO
      LEFT=0
```

```
TOTAL LENGTH OF L-TH STRIP

DX=2.DO*BOX(L)
DO 3 J=1,IDIV1(L)
DO 2 I=1,N
ITEST(I)=0
XMIN=XR(I)-BOX(L)
U=RANF()*DX+XMIN
X(I)=U*TSTD(I)+TMEAN(I)
IF(L.EQ.ISTRIP) GO TO 2
IF(U.GE.XR(I)-BOX(L+1).AND.U.LE.XR(I)+BOX(L+1)) ITEST(I)=1
CONTINUE
IF(L.EQ.ISTRIP) GO TO 230
ITE=0
DO 21 I=1,N
ITE=ITE+ITEST(I)
CONTINUE
IF(ITE.EQ.N) THEN
   LEFT=LEFT+1
ELSE
   IDIV(L)=IDIV(L)+1
END IF
IF(ITE.EQ.N) GO TO 3
IF(G(X).GT.0.DO) GO TO 3

SUMX = SUM OF f(x1,x2,...,xn)

SUM1=1.DO
DO 5 I=1,N
SUM1=SUM1*F(IDIST(I),XMEAN(I),STD(I),X(I),AL(I),BE(I))
CONTINUE
SUMX=SUMX+SUM1
CONTINUE

SUM2 = PRODUCT OF EACH STD

SUM2=1.DO
DO 6 I=1,N
SUM2=SUM2*TSTD(I)
CONTINUE
IF(L.EQ.ISTRIP) IDIV(L)=IDIV1(L)
SUMX=SUMX*AREA(L)*SUM2/IDIV(L)
SUM=SUM+SUMX
IDIV1(L+1)=IBOX(L+1)+LEFT
CONTINUE
BETA=-XINV(SUM)
SUMP=SUMP+SUM
SUMS=SUMS+SUM**2
CALL SECOND(TX2)
TIME=TX2-TX1
WRITE(6,771) SUM,BETA,TIME
CONTINUE
SUMS=DSQRT((SUMS-SUMP**2/NT)/(NT-1.DO))
SUMP=SUMP/NT
WRITE(6,772) SUMP
WRITE(6,773) SUMS
FORMAT(/,1X,'INITIAL STARTING POINT (REDUCED VARIATES)')
FORMAT(/,1X,'NEW STARTING POINT (REDUCED VARIATES)')
FORMAT(5(1X,F6.3))
```

```
764     FORMAT(1X,'SUM(',I2,') = ',1PE12.4)
765     FORMAT(1X,'GAMMA(',I2,') = ',I5)
766     FORMAT(1X,'K = ',I5)
7662    FORMAT(/,1X,'STRIP (SIGMA) =',5(1X,F6.2))
7661    FORMAT(/,1X,'SAMPLE (POINTS) =',5(1X,I5))
767     FORMAT(//)
768     FORMAT(2X,A7,2X,A7,1X,1PE12.4,1PE12.4,/)
769     FORMAT(//,1X,'VARIABLE',1X,' DIST. ',6X,'MEAN',8X,'STD. DEV',/)
770     FORMAT(//,1X,'ESTIMATE OF I',4X,'BETA',4X,'CPU SEC',/)
771     FORMAT(2X,1PE12.4,3X,0PF6.3,2X,0PF6.2,/)
772     FORMAT(1X,'AVG. OF ESTIMATION = ',1PE12.4,/)
773     FORMAT(1X,'STANDARD DEVIATION = ',1PE12.4,/)
        STOP
        END


        DOUBLE PRECISION FUNCTION F(IDIST,XMEAN,XDEV,X,ALPHA,BETA)

C       CACULATE PDF OF EACH VARIABLE

        IMPLICIT DOUBLE PRECISION (A-H,O-Z)
        COMMON /TWO/ PI,PI2,SPI2
        GO TO (1,2,3,4,5),IDIST
1       IF(X.LE.1.D-10) THEN
        F=0.DO
        GO TO 10
        END IF
        RB=X/BETA
        EW=RB**ALPHA
        IF(EW.GT.200.) EW=200.
        EXPWEI=DEXP(-EW)
        F=(ALPHA/BETA)*(EW/RB)*EXPWEI
        GO TO 10
    2   Z=(X-XMEAN)/XDEV
        F=SPI2*DEXP(-Z**2*.5D0)/XDEV
        GO TO 10
    3   EE=ALPHA*(X-BETA)
        IF(EE.GT.200.) EE=200.
        YY=DEXP(-EE)
        IF(YY.GT.200.) YY=200.
        EY=EE+YY
        IF(EY.GT.200.) EY=200.
        F=ALPHA*DEXP(-EY)
        GO TO 10
4       IF(X.LE.0.DO) THEN
        F=0.DO
        GO TO 10
        END IF
        CX21=XDEV**2+1.DO
        YMEAN=DLOG(XMEAN)
        YDEV=DSQRT(DLOG(CX21))
        Z=(DLOG(X)-YMEAN)/YDEV
        EZ=-(Z**2)*.5DO
        IF(EZ.LE.-200.) EZ=-200.
        F=SPI2*DEXP(EZ)/(YDEV*X)
        GO TO 10
5       IF(X.LE.0.DO) THEN
        F=0.DO
        GO TO 10
```

```
      END IF
      TEMP=(BETA/X)**ALPHA
      IF(TEMP.GE.200.) TEMP=200.
      CDF=DEXP(-TEMP)
      F=CDF*TEMP*ALPHA/X
  ) RETURN
    END


      DOUBLE PRECISION FUNCTION CDFNOR(Z)
      THIS FUNCTION COMPUTES THE NORMAL CDF.
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      COMMON /TWO/ PI,PI2,SPI2
      DATA A/0.31938153D0/,B/-0.356563782D0/,C/1.78147937D0/,
     + D/-1.821255978D0/,E/1.33027442 9D0/
      EZ=-(Z**2)*.5D0
      CDFNOR=0.0D0
      IF(EZ.LE.-200.0D0) GO TO 1
      ZX=SPI2*DEXP(EZ)
      IF(DABS(Z).GT.6.D0) GO TO 2
      T=1.D0/(1.D0+(0.2316419D0*DABS(Z)))
      CDFNOR=ZX*T*(A+T*(B+T*(C+T*(D+T*E))))
      GO TO 1
      Z2=1.D0/(Z*Z)
      CDFNOR=ZX*(1.D0-Z2*(1.D0-3.D0*Z2*(1.D0-5.D0*Z2)))/DABS(Z)
      IF(Z.GT.0.0D0) CDFNOR=1.0D0-CDFNOR
      RETURN
      END


      DOUBLE PRECISION FUNCTION XINV (Z)

      INVERSE NORMAL CDF

      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      F(X,P1)=P1-CDFNOR(X)
      Y=Z
      IF(Z.GT.0.5D0) Y=1.D0-Z
      C0=2.515517D0
      C1=0.802853D0
      C2=0.010328D0
      D1=1.432788D0
      D2=0.189269D0
      D3=0.001308D0
      T=(-2.D0*DLOG(Y))**.5D0
      DNUM=C0+T*(C1+T*C2)
      DNOM=1.0D0+T*(D1+T*(D2+T*D3))
      X=T-(DNUM/DNOM)
      IF(Z.LT.0.5D0) X=-X
      X1=X
      F1=F(X1,Z)
      X2=X1+.001D0
      F2=F(X2,Z)
      XX=X2
      CONTINUE
      IF(DABS(XX-X1).GE.1.D-10) THEN
      XX=X2-F2*(X2-X1)/(F2-F1)
      X1=X2
      X2=XX
```

```
      F1=F2
      F2=F(XX,Z)
      GO TO 10
      END IF
      XINV=XX
      END


      SUBROUTINE SECT1(COV,ISIGN,ALPHA)

C     CALCULATE ALPHA, AND BETA IN WEIBULL OR FRECHET
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      COMMON /TWO/ PI,PI2,SPI2

C     ISIGN = 1; WEIBULL DIST.
C           = 2; FRECHET DIST.

      F(X,COV)=-(1.DO+COV**2)*GAMMA(X)**2+GAMMA(2.*X)
      IF(ISIGN.EQ.1) X1=COV**(1.08)
      IF(ISIGN.EQ.2) X1=COV**(.677)/2.33
      IF(ISIGN.EQ.2.AND.X1.GT..49DO) X1=.48999999
7     IF(ISIGN.EQ.1) F1=F(X1,COV)
      IF(ISIGN.EQ.2) F1=F(-X1,COV)
      IF(DABS(F1).LE.1.D-10) GO TO 1
      X2=X1+.01DO
      IF(ISIGN.EQ.1) F2=F(X2,COV)
      IF(ISIGN.EQ.2) F2=F(-X2,COV)
      XX=X2
10    CONTINUE
      IF(DABS(XX-X1).GE.1.D-9) THEN
      XX=X2-F2*(X2-X1)/(F2-F1)
      X1=X2
      X2=XX
      F1=F2
      IF(ISIGN.EQ.1) F2=F(XX,COV)
      IF(ISIGN.EQ.2) F2=F(-XX,COV)
      GO TO 10
      END IF
      X1=XX
1     ALPHA=1.DO/X1
      RETURN
      END


      SUBROUTINE WEI(XMEAN,XDEV,ALPHA,BETA)

C     CALCULATE PARAMETERS (ALPHA AND BETA)

      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      COV=XDEV/XMEAN
      CALL SECT1(COV,1,ALPHA)
      AL1=1.DO/ALPHA
      BETA=XMEAN/GAMMA(AL1)
      RETURN
      END


      SUBROUTINE FRE(XMEAN,XDEV,ALPHA,BETA)
```

```
CALCULATE PARAMETERS (ALPHA AND BETA)

IMPLICIT DOUBLE PRECISION (A-H,O-Z)
COV=XDEV/XMEAN
CALL SECT1(COV,2,ALPHA)
AL1=1.DO/ALPHA
BETA=XMEAN/GAMMA(-AL1)
RETURN
END


SUBROUTINE EVD(XMEAN,STD,ALPHA,BETA,PI)

CALCULATE PARAMETERS (ALPHA AND BETA)

IMPLICIT DOUBLE PRECISION (A-H,O-Z)
ALPHA=PI/(STD*DSQRT(6.DO))
BETA=XMEAN-.57721566490153/ALPHA
RETURN
END


DOUBLE PRECISION FUNCTION GAMMA(Y1)

GAMMA FUNCTION

IMPLICIT DOUBLE PRECISION (A-H,O-Z)
COMMON /TWO/ PI,PI2,SPI2
X=Y1+1.D+0
Z=X
IF(X.GE.6.0D+0)GO TO 456
N=INT(X)
Z=(6.0D+0)-N+X
Y=1.D+0/Z**2
ALG=(Z-.5D+0)*DLOG(Z)+.5D+0*DLOG(PI2)-
Z-(1.D+0/(12.D+0*Z))*(((Y/0.14D+3-1.D+0/0.105D+3)*Y+
1.D+0/.3D+2)*Y-1.D+0)
IF(X.GE.6.D+0)GO TO 457
ITE=6-N
DO 3 J=1,ITE
A=X+J-1.D+0
ALG=ALG-DLOG(A)
CONTINUE
GAMMA=DEXP(ALG)
RETURN
END


SUBROUTINE RACA(Z,N,XR,EPS,XMEAN,IDIST,XDEV,XCOV,AL,BE,BETA,ZO)

SUBROUTINE FOR H-L OR R-F

IMPLICIT DOUBLE PRECISION (A-H,O-Z)
DIMENSION X(20),XR(20),BUFFER(20),D(20),XMEAN(20),XDEV(20),
IDIST(20),XNMEAN(20),XNDEV(20),AL(20),BE(20),Z(20),XCOV(20)
COMMON /DIREC/ DG(20)
COMMON /RACAXX/ ZO1
COMMON /RAC/ LL
```

```
*       OPTIMIZATION ROUTINE BEGINS HERE
        ZO1=ZO

        KK=0
        CALL HAZL(Z,N,F,EPS,D,XR,XMEAN,XDEV,IDIST,X,BETA,KK)
        IF(LL.EQ.0) RETURN

*       THIS LOOP CALCULATES THE EQUIVALENT NORMAL DISTRIBUTION
*       FOR EACH DESIGN  VARIABLE.

 17     KK=KK+1
        DO 13 J=1,N
        CALL FIND(AL(J),BE(J),IDIST(J),XMEAN(J),XDEV(J),X(J),
     $  XNMEAN(J),XNDEV(J))
 13     CONTINUE
        ZBETA=BETA
        CALL HAZL(Z,N,F,EPS,D,XR,XNMEAN,XNDEV,IDIST,X,BETA,KK)

C*      MAX OF 35 ITERATIONS FOR EQUIVALENT NORMAL SEARCH ALGORITHM
C
        IF(KK.EQ.35) GO TO 19
   21   IF(ABS(BETA-ZBETA).LE.0.0001) GO TO 19
        GO TO 17
19      RETURN
        END


C
C*      HASOFER-LIND SAFETY INDEX CALCULATIONS
C
        SUBROUTINE HAZL(Z,N,F,T,D,XR,XMEAN,XDEV,IDIST,X,BETA,KK)
        IMPLICIT DOUBLE PRECISION (A-H,O-Z)
        DIMENSION X(20),XR(20),BUFFER(20),D(20),XMEAN(20),XDEV(20)
     +, IDIST(20),Z(20)
        IF(KK.EQ.10) T=0.1*T
        IF(KK.EQ.20) T=0.1*T
        IF(KK.EQ.25) T=0.1*T
        IF(KK.EQ.34) T=0.1*T
C       ININTIAL GUESS XR
        EPSI=1.E-4
        IF(KK.EQ.0) CALL GFN(EPSI,IDIST,XMEAN,XDEV,Z,N,XR,0)
        DO 1 I=2,N
        D(I)=0.1
   1    CONTINUE
C
C*      OPTIMIZATION ROUTINE BEGINS HERE
C
        CALL FN(Z,N,F,X,XR,XMEAN,XDEV,IDIST)
        BETA=F
        DO 3 N1=2,N
    3   BUFFER(N1)=XR(N1)
   15   DO 4 N1=2,N
        DO 5 N4=2,3
        XR(N1)=XR(N1)+D(N1)*(-1.)**N4
        CALL FN(Z,N,F,X,XR,XMEAN,XDEV,IDIST)
        IF(F.GE.BETA) GO TO 45
        BUFFER(N1)=XR(N1)
        M1=0
   25   BETA=F
```

```
XR(N1)=XR(N1)+D(N1)*(-1.)**N4
CALL FN(Z,N,F,X,XR,XMEAN,XDEV,IDIST)
IF(F.LT.BETA) GO TO 35
XR(N1)=BUFFER(N1)
GO TO 55
M1=M1+1
BUFFER(N1)=XR(N1)
IF(M1.LT.3) GO TO 25
DO 6 N2=2,N
D(N2)=D(N2)*2.
GO TO 55
XR(N1)=BUFFER(N1)
CONTINUE
CONTINUE
DO 7 N1=2,N
D(N1)=D(N1)*.5
CONTINUE
DO 8 I=2,N
IF(D(I).GE.T) GO TO 15
CONTINUE
RETURN
END


SUBROUTINE FN(Z,N,F,X,XR,XMEAN,XDEV,IDIST)

OPTIMIZATION SUBROUTINE

IMPLICIT DOUBLE PRECISION (A-H,O-Z)
DIMENSION X(20),XR(20),XMEAN(20),XDEV(20),IDIST(20),Z(20)
COMMON /RACAXX/ ZO1
SUM=0.0

COMPUTATION OF BASIC VARIABLES FROM GUESS OF REDUCED VARIABLES

DO 1 I=2,N
X(I)=XDEV(I)*XR(I)+XMEAN(I)
CONTINUE
DO 2 I=2,N
IF(IDIST(I).NE.4)GO TO 2

RECOMPUTATION BACK TO BASIC FORM FOR LOG TRANSFORMED VARIABLES

X(I)=DEXP(X(I))
CONTINUE
CALL LSFRA(N,X,Z,ZO1)

COMPUTATION OF REDUCED VALUE OF DEPENDENT VARIABLE.  TRANSFORM IS MADE
IF SPECIFIED

IF(IDIST(1).NE.4)GO TO 15
IF(X(1).LE.1.D-20) X(1)=1.D-20
XR(1)=(DLOG(X(1))-XMEAN(1))/XDEV(1)
GO TO 25
XR(1)=(X(1)-XMEAN(1))/XDEV(1)
CONTINUE

CALCULATION OF BETA, THE SAFETY INDEX
```

```
      DO 3 I=1,N
      IF(XR(I).GT.27.) XR(I)=27.
      IF(XR(I).LT.-27.) XR(I)=-27.
      SUM=SUM+XR(I)**2
    3 CONTINUE
      F=DSQRT(SUM)
      RETURN
      END


      SUBROUTINE FIND(ALPHA,BETA,IDIST,XMEAN,XDEV,X,XNMEAN,XNDEV)
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      COMMON /TWO/ PI,PI2,SPI2
      GO TO (1,4,3,4,5),IDIST
      RB=X/BETA
      EW=RB**ALPHA
      IF(EW.GT.200.) EW=200.
      EXPWEI=DEXP(-EW)
      CDF=1.DO-EXPWEI
      PDF=(ALPHA/BETA)*(EW/RB)*EXPWEI
      GO TO 20
    3 EE=ALPHA*(X-BETA)
      IF(EE.GT.200.) EE=200.
      YY=DEXP(-EE)
      IF(YY.GT.200.) YY=200.
      CDF=DEXP(-YY)
      EY=EE+YY
      IF(EY.GT.200.) EY=200.
      PDF=ALPHA*DEXP(-EY)
      GO TO 20
4     XNMEAN=XMEAN
      XNDEV=XDEV
      GO TO 10
5     TEMP=(BETA/X)**ALPHA
      IF(TEMP.GT.200.) TEMP=200.
      CDF=DEXP(-TEMP)
      PDF=CDF*TEMP*ALPHA/X
C     R-F TRANSFORMATION
20    PDFNOR=SPI2*DEXP(-(XINV(CDF)**2)*.5E0)
      XNDEV=PDFNOR/PDF
      XNMEAN=X-XINV(CDF)*XNDEV
   10 RETURN
      END


      SUBROUTINE GFN(EPSI1,IDI,XM,ST,Z,N,XR,KK)

C     FIRST INITIAL GUESS FOR XR

      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      DIMENSION XM(20),STD(20),IDI(20),XR(20),Z(20),DIR(20)
      DIMENSION ST(20),XT(20)
      COMMON /DIREC/ DG(20)
      DO 10 I=1,N
      XT(I)=XM(I)
      STD(I)=ST(I)
      IF(KK.EQ.0.AND.IDI(I).EQ.4) THEN
      XT(I)=DEXP(XM(I)+.5*ST(I)**2)
      TEMP=DEXP(ST(I)**2)
```

```
STD(I)=DSQRT(DEXP(2.*XM(I))*TEMP*(TEMP-1.))
GO TO 10
END IF
CONTINUE
GBAR=G(XT)
DO 2 I=1,N
EPSI=EPSI1
IF(STD(I).LT.1.) EPSI=STD(I)*1.D-4
XT(I)=XT(I)+EPSI
GXT=G(XT)
DG(I)=(GXT-GBAR)/EPSI*STD(I)
XT(I)=XT(I)-EPSI
CONTINUE
SUM=0.0
DO 3 I=1,N
SUM=SUM+DG(I)**2
CONTINUE
DSUM=DSQRT(SUM)
DO 5 I=1,N
DIR(I)=DG(I)/DSUM
CONTINUE
BETA=GBAR/DSUM
DO 6 I=1,N
XR(I)=-DIR(I)*BETA
CONTINUE
RETURN
END
```

```
USER SUPLLIED SUBROUTINES

SUBROUTINE LSFRA(N,X,Z,ZO)

SUBROUTINE FOR LIMIT STATE FUNCTION

REQUIRED BY H-L OR R-F
USE Z(20),ZO FOR CONSTANT VALUES

IMPLICIT DOUBLE PRECISION (A-H,O-Z)
DIMENSION X(N),Z(20)

LIMIT STATE FUNCTION (X(1)=f(X(2),X(3),...,X(N))

X(1)=DSQRT(300.*X(2)**2+1.92*X(3)**2)
RETURN
END
```

Response functions for the examples given

```
DOUBLE PRECISION FUNCTION G(X)

FUNCTION FOR Monte Carlo

IMPLICIT DOUBLE PRECISION (A-H,O-Z)
COMMON /TWO/ PI,PI2,SPI2
DIMENSION X(20)

PERFORMANCE FUNCTION (LESS THAN OR EQAUL TO ZERO TYPE)

G=X(1)-DSQRT(300.*X(2)**2+1.92*X(3)**2)

RETURN
END
```

Appendix F. Listing of the Harbitz Program

This program was developed to run on the VAX and the CYBER 175. The listing fiven here is for the CYBER version. The VAX version runs in double precision. It is not interactive.

The performance function $g(\underset{\sim}{X})$ must be introduced in two subroutines.

1) Subroutine HARBIFN. Enter the function $g(\underset{\sim}{X})$ directly. See the listing for an example.

2) Subroutine LSFFPI. Here the limit state $g(\underset{\sim}{X}) = 0$ is entered such that one variable is a function of the others. See the listing for an example.

The reason that $g(\underset{\sim}{X})$ must be entered in two places in a different format has to do with the calculational procedure. The Rackwitz-Fiessler algorithm to perform the first order reliability enelysis uses an optimization routine and required that the limit state be entered. A significant improvement to the program would result if a R-F routine requiring $g(\underset{\sim}{X})$ as input be implimented.

Data Input File

Card 1 Problem identification in "A" format

Card 2 EPS, K, N, Z0

EPS; The stop criterion for FPI

K; number of random variables

N; number of trials

Z0; constant used for constructing cdf, e.g., $p = P[h(\underset{\sim}{X}) - Z0]$

Define $g(\underset{\sim}{X}) = h(\underset{\sim}{X}) - Z0$

It is most convenient to change Z0 through the data than it is a Fortran statement.

Cards 3 and 4 are repeated for each variable.

Card 3 Variable name in "Z" format

Card 4 DIST(I), XMEAN(I), STD(I)

DIST(I) = 1 WEIBULL

2 NORMAL

3 EVD

4 LOGNORMAL

5 FRECHET

XMEAN(I) = mean value; median if lognormal

STD(I) = standard deviation; COV if lognormal

```
PROGRAM HARBITZ(INPUT,OUTPUT,TAPE5=INPUT,TAPE6=OUTPUT)
IMPLICIT REAL (A-H,O-Z)
CHARACTER*80 ANS
CHARACTER VNAME(20)*5,VDIST(20)*7

UP TO 20 RANDOM VARIABLES

DIMENSION X(20),XRAN(20),ZX(20),XR(20)
DIMENSION DIST(20),XMEAN(20),XCOV(20),STD(20),AL(20),BE(20)
COMMON /TWO/ PI,PI2,SPI2
COMMON /ASA/ AX

EQ. 22 IN HARBITZ'S PAPER

G(U,XNU,ALPHA)=(-LOG(U))**(XNU*.5EO-1.EO)*U**(.5EO*ALPHA-1.EO)

START PROGRAM

ISEED=TIME(DUMMY)
CALL RANSET(ISEED)

CACULATE CONSTANT PARAMETERS

PI=4.EO*ATAN(1.EO)
PI2=PI+PI
SPI2=1.EO/SQRT(PI2)

READ INPUT DATA

READ(5,'(A)') ANS

ANS IS USED FOR THE PROBLEM IDENTIFICATION

READ(5,*) EPS,K,N,ZO

EPS IS USED IN FPI FOR STOP CRITERION
ISEED IS INITIAL SEED NUMBER FOR EANDOM NUMBER GENERATION
K IS NUMBER OF RANDOM VARIABLES
N IS NUMBER OF TRIALS

DO 610 I=1,K

DIST(I) = 1.; WEIBULL
        = 2.; NORMAL
        = 3.; EVD
        = 4.; LOGNORMAL
        = 5.; FRECHET
READ(5,'(A)') VNAME(I)
READ(5,*) DIST(I),XMEAN(I),STD(I)
IF(DIST(I).EQ.1.) VDIST(I)='WEIBULL'
IF(DIST(I).EQ.2.) VDIST(I)='NORMAL'
IF(DIST(I).EQ.3.) VDIST(I)='EVD'
IF(DIST(I).EQ.4.) VDIST(I)='LOG'
IF(DIST(I).EQ.5.) VDIST(I)='FRECHET'

IF LOGNORMAL, USE MEDIAN, AND COV

GO TO (601,600,603,604,605), INT(DIST(I)+.1EO)
```

Note that the random process is initiated using the clock.

Program HARBITZ Monte Carlo using the Harbitz method. This version runs on the CYBER. The VAX version is in double precision.

```
      CALCULATE DISTRIBUTION PARAMETERS
C
601   CALL WEI(XMEAN(I),STD(I),AL(I),BE(I))
      GO TO 600
603   CALL EVD(XMEAN(I),STD(I),AL(I),BE(I),PI)
      GO TO 600
604   XMEAN(I)=XMEAN(I)*SQRT(1.E0+STD(I)**2)
      STD(I)=XMEAN(I)*STD(I)
      GO TO 600
605   CALL FRE(XMEAN(I),STD(I),AL(I),BE(I))
600   CONTINUE
      XCOV(I)=STD(I)/XMEAN(I)
610   CONTINUE
C
C     XNU IS DEGREES OF FREEDOM IN CHI-SQUARED DISTRIBUTION
C
      XNU=REAL(K)
C
C     IR=0 FOR USING EQ. 25 IN HARBITZ'S PAPER
C     IR=1 FOR USING EQ. 26 IN HARBITZ'S PAPER
C
      IR=0
C
C     START TO CHECK CPU TIME CONSUMED
C
      CALL SECOND(TX1)
C
C     CALL XFPI TO CALCULATE BETA
C     ZX, AND ZO CAN BE USED FOR CONSTANTS
C
      CALL XFPI(ZX,K,XR,EPS,XMEAN,DIST,STD,XCOV,AL,BE,BETA,ZO)
C
C     IF K=2, THEN ALPHA IS ALWAYS 2.0, AND AR IS 1.0
C
      IF(K.EQ.2) THEN
         AR=1.E0
         ALPHA=2.E0
         GO TO 230
      END IF
C
C     STARTING MIN. ALPHA = 2.0
C     STARTING MAX. ALPHA = 10.
C
      ALMIN1=2.E0
      ALMAX1=10.E0
C
C     IT IS USED FOR MAX. ALPHA IS G.T. 10.0
C
      IT=0
220   CONTINUE
      ALMIN=ALMIN1
      ALMAX=ALMAX1
C
C     MORE ACCURACY IS NEEDED, INCREASE ITERATION NUMBER (e.g., 5 OR 6)
C
      DO 200 I=1,3
C
C     FIND ALPHA FOR MAX. AREA RATIO, AR
C
```

```
CALL HATEST(ALMIN,ALMAX,BETA,K,ALMI,ALMA,ALPHA,AR)

MAX. ALPHA GOES TO THE RIGHT HAND SIDE (G.T. 20., OR 30.)

IF(ALPHA.EQ.ALMAX1) THEN
   IT=IT+1

MAX. ALPHA IS SET TO BE 50.

   IF(IT.EQ.6) GO TO 230
   ALMIN1=REAL(IT)*10.EO
   ALMAX1=REAL(IT+1)*10.EO
   GO TO 220
END IF
ALMIN=ALMI
ALMAX=ALMA
CONTINUE
CONTINUE

EQ.21 IN HARBITZ'S PAPER

BETA3=BETA+3.EO
U1=EXP(-(BETA3)**2/ALPHA)
U2=EXP(-BETA**2/ALPHA)

CALCULATE CONSTANT PARAMETERS FIRST

U12=U2-U1

NUM IS TOTAL NUMBER OF G<O
NR IS TOTAL NUMBER OF RADIUS CALCULATION (EQ. 24, AND 25)

NUM=O
NR=O

EQ.23 IN HARBITZ'S PAPER

IF(XNU.EQ.2.EO) THEN

ARG IS Umax

   ARG=(U1+U2)*.5EO
   GARG=1.EO
ELSE
   ARG=EXP(-(XNU-2.EO)/(ALPHA-2.EO))
   GARG=G(ARG,XNU,ALPHA)
END IF

FIND Gmax

IF(ARG.GE.U1.AND.ARG.LE.U2) THEN

Umax IS BETWEEN U1 AND U2

GX=GARG
IGMAX=O
ELSE

Umax IS U1 OR U2
```

```
      GU1=G(U1,XNU,ALPHA)
      GU2=G(U2,XNU,ALPHA)
      GX=AMAX1(GU1,GU2)
      IGMAX=1
      END IF
      DO 1 I=1,N

      SAMPLE UNTIL G<g(U)

0     UJ=RANF()*U12+U1
      GJ=RANF()*GX

      UJ IS SAMPLED BETWEEN U1 AND U2
      GJ IS SAMPLED BETWEEN 0 AND g(Umax)

      NR=NR+1
      IF(GJ.GE.G(UJ,XNU,ALPHA)) GO TO 10
      IF(IR.EQ.0) THEN

      EQ. 25 IN HARBITZ'S PAPER

         RJ=SQRT(-ALPHA*LOG(UJ))
      ELSE

      EQ. 26 IN HARBITZ'S PAPER

         L1=MOD(K,2)
         K2=K/2
         IF(L1.EQ.1) K2=(K-1)/2
100      CONTINUE
         NR=NR+1
         SUMR=1.E0
         DO 110 IXY=1,K2
         SUMR=SUMR*RANF()
110   CONTINUE
         IF(L1.EQ.0) THEN

      EVEN NUMBER  ANDOM VARIABLES (EQ. 26 A)

            RJ=-2.E0*LOG(SUMR)
         ELSE

      ODD NUMBER RANDOM VARIABLES (EQ. 26b)

            X1=-2.E0*LOG(RANF())
            X2=RANF()*PI2
            X3=SQRT(X1)*COS(X2)
            RJ=-2.E0*LOG(SUMR)+X3**2
         END IF
      END IF
      SUM=0.E0

      GENERATE STANDARD NORMAL VARIATES

      DO 2 J=1,K
      X1=-2.E0*LOG(RANF())
      X2=RANF()*PI2
      X(J)=SQRT(X1)*COS(X2)
```

```
SUM=SUM+X(J)**2
CONTINUE
SUM=SQRT(SUM)

NORMALIZATION OF NORMAL VARIATES (EQ. 27 IN HARBITZ'S PAPER)

DO 3 J=1,K
X(J)=X(J)/SUM*RJ
CONTINUE

INVERSE TRANSFORMATION FROM NORMAL VARIATE

DO 500 J=1,K
GO TO (501,502,503,504,505), INT(DIST(J)+.1E0)

WEIBULL DISTRIBUTION

XRAN(J)=BE(J)*(-LOG(1.E0-CDFNOR(X(J))))**(1.E0/AL(J))
GO TO 500

NORMAL DISTRIBUTION

XRAN(J)=STD(J)*X(J)+XMEAN(J)
GO TO 500

EVD

XRAN(J)=BE(J)-LOG(-LOG(CDFNOR(X(J))))/AL(J)
GO TO 500

LOGNORMAL DISTRIBUTION

CX2=1.E0+XCOV(J)**2
XMEANJ=LOG(XMEAN(J)/SQRT(CX2))
STDJ=SQRT(LOG(CX2))
XRAN(J)=EXP(STDJ*X(J)+XMEANJ)
GO TO 500

FRECHET DISTRIBUTION

XRAN(J)=BE(J)*(-LOG(CDFNOR(X(J))))**(-1.E0/AL(J))
CONTINUE

PERFORMANCE FUNCTION

CALL HARBIFN(XRAN,K,ALPHA,BETA,Z)

FIND TOTAL NUMBER OF G < 0

IF(Z.LT.0.E0) NUM=NUM+1
CONTINUE
B2=BETA**2

CHIX IS PROBABILITY IN BETA SPHERE

CHIX=1.E0-CHI(B2,XNU)

PF IS PROBABILITY OF FAILURE
```

```
      PRINT INPUT DATA

      PFZ=REAL(NUM)/REAL(N)
      PF=CHIX*PFZ
      XPF=-XINV(PF)
      CLT=XINV(.975)*SQRT(PFZ*(1.E0-PFZ)/DBLE(N))/PFZ
      CL=PF*(1.E0-CLT)
      IF(CL.LT.0.E0) CL=0.E0
      UL=PF*(1.E0+CLT)
      CALL SECOND(TX2)
      DTT=TX2-TX1
      WRITE(6,'(///,9X,A)') ANS
      WRITE(6,910)
910   FORMAT(///,30X,'DESIGN VARIABLES',//,9X,'VARIABLE',7X,
     +' DISTRIBUTION',9X,'MEAN/MEDIAN',8X,' STD/COV')
      DO 781 I=1,K
      IF(DIST(I).EQ.4.) THEN
         XMEAN(I)=XMEAN(I)/SQRT(1.E0+XCOV(I)**2)
         STD(I)=XCOV(I)
      END IF
      WRITE(6,920) VNAME(I),VDIST(I),XMEAN(I),STD(I)
920   FORMAT(/,10X,A5,12X,A7,10X,1PE12.4,7X,1PE12.4)
781   CONTINUE
      WRITE(6,'(/,8X,A,/)') ' (NOTE: THE MEDIAN AND COV USED FOR LN)'
      WRITE(6,'(8X,A,F7.3,/)') ' BETA (SPHERE) = ',BETA
      WRITE(6,'(8X,A,I3,/)') ' NUMBER OF VARIABLES = ',K
      WRITE(6,'(8X,A,F7.4,/)') ' AREA RATIO, AR = ',AR
      WRITE(6,'(8X,A,F8.4,/)') ' ALPHA = ',ALPHA
      WRITE(6,'(8X,A,I5,/)') ' NUMBER OF SAMPLES = ',N
      WRITE(6,'(8X,A,I5,/)') ' TOTAL NUMBER OF g < 0 = ',NUM
      WRITE(6,'(8X,A,I5,/)') ' TOTAL NUMER OF POINTS SAMPLED = ',NR
      WRITE(6,'(8X,A,1PE13.5,/)') ' PROBABILITY IN BETA SPHERE = ', CHIX
      WRITE(6,'(8X,A,1PE13.5,/)') ' PROBABILITY OF FAILURE = ',PF
      WRITE(6,'(8X,A,F9.5,/)') ' BETA = ', XPF
      WRITE(6,'(8X,A,/)') ' 95 % CONFIDENCE INTERVAL ON PF'
      WRITE(6,'(8X,A,1PE13.5,/)') ' LOWER = ', CL
      WRITE(6,'(8X,A,1PE13.5,/)') ' UPPER = ', UL
      WRITE(6,'(/,8X,A,F8.2,/)') ' CPU EXECUTION TIME (SEC.) = ',DTT
      STOP
      END

C
C     CHI-SQUARED DISTRIBUTION FUNCTION
C
      REAL FUNCTION CHI(X,XNU)
      IMPLICIT REAL (A-H,O-Z)
C     REAL*16 DIV,RX
      SUM1=1.E0
      R=1.E0
      RX=X
      DIV=XNU+2.E0*R
1     CONTINUE
      SUM2=RX/DIV
      SUM1=SUM1+SUM2
      IF(SUM2.LE.1.E-10) GO TO 2
      RX=RX*X
      R=R+1.E0
      DIV=DIV*(XNU+2.E0*R)
      GO TO 1
```

```
   CONTINUE
   X2=.5E0*X
   XNU2=.5E0*XNU
   CHI=X2**XNU2*EXP(-X2)/GAMMA(XNU2)*SUM1
   RETURN
   END

   GAMMA FUNCTION

   REAL FUNCTION GAMMA(Y1)
   IMPLICIT REAL (A-H,O-Z)
   COMMON /TWO/ PI,PI2,SPI2
   X=Y1+1.E0
   Z=X
   IF(X.GE.6.0E0)GO TO 456
   N=INT(X)
   Z=(6.0E0)-N+X
   Y=1.E0/Z**2
   ALG=(Z-.5E0)*LOG(Z)+.5E0*LOG(PI2)-
$  Z-(1.E0/(12.E0*Z))*(((Y/140.E0-1.E0/105.E0)*Y+
$  1.E0/30.E0)*Y-1.E0)
   IF(X.GE.6.E0)GO TO 457
   ITE=6-N
   DO 3 J=1,ITE
   A=X+J-1.E0
   ALG=ALG-LOG(A)
   CONTINUE
   GAMMA=EXP(ALG)
   RETURN
   END

   STANDARD NORMAL CDF

   REAL FUNCTION CDFNOR(Z)
   THIS FUNCTION COMPUTES THE NORMAL CDF.
   IMPLICIT REAL (A-H,O-Z)
   COMMON /TWO/ PI,PI2,SPI2
   DATA A/0.31938153E0/,B/-0.356563782E0/,C/1.781477937E0/,
  +D/-1.821255978E0/,E/1.330274429E0/
   EZ=-(Z**2)*.5E0
   CDFNOR=0.0E0
   IF(EZ.LE.-200.0E0) GO TO 1
   ZX=SPI2*EXP(EZ)
   IF(ABS(Z).GT.6.E0) GO TO 2
   T=1.E0/(1.E0+(0.2316419E0*ABS(Z)))
   CDFNOR=ZX*T*(A+T*(B+T*(C+T*(D+T*E))))
   GO TO 1
   Z2=1.E0/(Z*Z)
   CDFNOR=ZX*(1.E0-Z2*(1.E0-3.E0*Z2*(1.E0-5.E0*Z2)))/ABS(Z)
   IF(Z.GT.0.0E0) CDFNOR=1.0E0-CDFNOR
   RETURN
   END

   INVERSE NORMAL CDF

   REAL FUNCTION XINV (Z)
   IMPLICIT REAL (A-H,O-Z)
   F(X,P1)=P1-CDFNOR(X)
   Y=Z
```

```
      IF(Z.GT.0.5E0) Y=1.E0-Z
      IF(Z.EQ.1.E0) STOP
      C0=2.515517E0
      C1=0.802853E0
      C2=0.010328E0
      D1=1.432788E0
      D2=0.189269E0
      D3=0.001308E0
      T=(-2.E0*LOG(Y))**.5E0
      DNUM=C0+T*(C1+T*C2)
      DNOM=1.0E0+T*(D1+T*(D2+T*D3))
      X=T-(DNUM/DNOM)
      IF(Z.LT.0.5E0) X=-X
      X1=X
      F1=F(X1,Z)
      X2=X1+.001E0
      F2=F(X2,Z)
      XX=X2
10    CONTINUE
      IF(ABS(XX-X1).GE.1.E-10) THEN
      XX=X2-F2*(X2-X1)/(F2-F1)
      X1=X2
      X2=XX
      F1=F2
      F2=F(XX,Z)
      GO TO 10
      END IF
      XINV=XX
      END
C
C     FIND PARAMETERS IN WEIBULL, OR FRECHET
C
      SUBROUTINE SECT1(COV,ISIGN,ALPHA)
      IMPLICIT REAL (A-H,O-Z)
      COMMON /TWO/ PI,PI2,SPI2
C     ISIGN = 1; WEIBULL DIST.
C           = 2; FRECHET DIST.
      F(X,COV)=-(1.E0+COV**2)*GAMMA(X)**2+GAMMA(2.*X)
      IF(ISIGN.EQ.1) X1=COV**(1.08)
      IF(ISIGN.EQ.2) X1=COV**(.677)/2.33
      IF(ISIGN.EQ.2.AND.X1.GT..49E0) X1=.48999999
7     IF(ISIGN.EQ.1) F1=F(X1,COV)
      IF(ISIGN.EQ.2) F1=F(-X1,COV)
      IF(ABS(F1).LE.1.E-10) GO TO 1
      X2=X1+.01E0
      IF(ISIGN.EQ.1) F2=F(X2,COV)
      IF(ISIGN.EQ.2) F2=F(-X2,COV)
      XX=X2
10    CONTINUE
      IF(ABS(XX-X1).GE.1.E-9) THEN
      XX=X2-F2*(X2-X1)/(F2-F1)
      X1=X2
      X2=XX
      F1=F2
      IF(ISIGN.EQ.1) F2=F(XX,COV)
      IF(ISIGN.EQ.2) F2=F(-XX,COV)
      GO TO 10
      END IF
      X1=XX
```

```
ALPHA=1.E0/X1
RETURN
END

PARAMETERS CALCULATION (ALPHA, AND BETA)

SUBROUTINE WEI(XMEAN,XDEV,ALPHA,BETA)
IMPLICIT REAL (A-H,O-Z)
COV=XDEV/XMEAN
CALL SECT1(COV,1,ALPHA)
AL1=1.E0/ALPHA
BETA=XMEAN/GAMMA(AL1)
RETURN
END

PARAMETERS CALCULATION (ALPHA, AND BETA)

SUBROUTINE FRE(XMEAN,XDEV,ALPHA,BETA)
IMPLICIT REAL (A-H,O-Z)
COV=XDEV/XMEAN
CALL SECT1(COV,2,ALPHA)
AL1=1.E0/ALPHA
BETA=XMEAN/GAMMA(-AL1)
RETURN
END

PARAMETERS CALCULATION (ALPHA, AND BETA)

SUBROUTINE EVD(XMEAN,STD,ALPHA,BETA,PI)
IMPLICIT REAL (A-H,O-Z)
ALPHA=PI/(STD*SQRT(6.E0))
BETA=XMEAN-.57721566490153/ALPHA
RETURN
END

FIND THE ALPHA FOR MAX. AREA RATIO

SUBROUTINE HATEST(ALMIN,ALMAX,BETA,K,ALMI,ALM' ALPA,AR1)
IMPLICIT REAL (A-H,O-Z)
DIMENSION G(21),AR(21)

20 SEGMENTS BETWEEN MIN. ALPHA AND MAX. ALPHA

DAL=(ALMAX-ALMIN)*.05E0
BET3=BETA+3.E0
XK2=DBLE(K)*.5E0
DO 1 IX=1,21
ALPHA=ALMIN+DBLE(IX-1)*DAL
AL2=ALPHA*.5E0
U1=EXP(-BET3**2/ALPHA)
U2=EXP(-BETA**2/ALPHA)
DU=(U2-U1)*.05E0
U12=U1+DU*.5E0
Gmax=0.E0
SUM=0.E0
DO 2 J=1,20
U=U12+(J-1)*DU

EQ. 22 IN HARBITZ'S PAPER
```

```
      G(J)=(-LOG(U))**(XK2-1.E0)*U**(AL2-1.E0)
      IF(G(J).GE.Gmax) Gmax=G(J)
      SUM=SUM+G(J)
      CONTINUE
      AR(IX)=SUM/(20.E0*Gmax)
      CONTINUE

C     FIND MAX. AR
C
      ARMAX=AR(1)
      DO 10 J=2,21
      IF(ARMAX.LE.AR(J)) THEN
         ARMAX=AR(J)
         IJ=J
      END IF
10    CONTINUE
      ALMI=ALMIN+DBLE((IJ-2))*DAL
      ALPA=ALMIN+DBLE((IJ-1))*DAL
      ALMA=ALMIN+DBLE(IJ)*DAL
      AR1=AR(IJ)
      RETURN
      END
      SUBROUTINE XFPI(ZX,N,XRZ,EPS,XMEAN,DIST,XDEV,XCOV,AL,BE,ZBET,ZO)
      IMPLICIT REAL (A-H,O-Z)
      DIMENSION AL(20),BE(20),ZX(20),X(20),DIST1(20),XRZ(20)
      DIMENSION XCOV(20),TXMEAN(20),TXCOV(20),XNMEAN(20),XNDEV(20)
      DIMENSION DIST(20),DX(20),XR(20),XMEAN(20),XDEV(20),TEMPXR(20)
      DIMENSION CI(20),AI(20),SF(20),STOREX(20),C(20,2),FORM(20)
      COMMON /OP1/ DIST1,DX,XR,XNMEAN,XNDEV,CI,AI,SF,C,FORM,II,ZXO,XR1
      COMMON /TWO/ PI,PI2,SPI2
C READ NUMBER OF VARIABLES(N), LIMIT STATE DESCRIPTION ('A' FORMAT).
C
C READ NAME, MEAN (MEDIAN FOR LOGNORMAL VARIABLES),COEFF. OF VARIATION,
C AND DISTRIBUTION TYPE (DEFINED IN SUB. CDFPDF) OF EACH VARIABLE.
      ZXO=ZO
      DO 15 I=1,N
      DIST1(I)=DIST(I)
      SF(I)=1.
15    CONTINUE
      KK=0
      II=0
      CO=0.
      AI(1)=0.
C READ ZO VALUE IN THE LIMIT STATE(DEFINED IN SUB. GFUNC): G(X)=Z(X)-ZO
C     WRITE(6,121)
C COMPUTE R-F BETA AND THE DESIGN POINT.----II=0 LOOP.
      CALL FIT(EPS,ZX,AL,BE,N,XMEAN,XDEV,ZBET,KK,II)
      RETURN
      END



      SUBROUTINE FIT(EPS,ZX,AL,BE,N,XMEAN,XDEV,BETA,KK,LL)
C THIS SUB. USES THE R-F ALGORITHM TO FIND THE R-F SAFETY INDEX. IT ALSO
C CONTROLS THE PROCESS OF CONSTRUCTING THREE PARAMETER EQUIV. NORMALS.
      IMPLICIT REAL (A-H,O-Z)
      DIMENSION DIST(20),DX(20),XR(20),XMEAN(20),XDEV(20),ZX(20)
      DIMENSION CI(20),AI(20),SF(20),XNMEAN(20),XNDEV(20),EE(20)
```

```
  DIMENSION C(20,2),FORM(20),AL(20),BE(20)
  COMMON /OP1/ DIST,DX,XR,XNMEAN,XNDEV,CI,AI,SF,C,FORM,II,ZO,XR1
    COMMON /TWO/ PI,PI2,SPI2
  II=LL
  IF(LL.NE.0) GO TO 40
MPUTE MINIMUM DISTANCE (SAFETY INDEX) USING SUBROUTINE OPTM.
SET INITIAL DESIGN POINT SEARCH VALUES (XR(I)).**
SET CONVERGENCE LIMITS (EE(I))  AND STEP SIZE MULTIPLIER (DD).**
  DO 30 I=1,N
  XR(I)=0.0
  EE(I)=0.0001
  XNMEAN(I)=XMEAN(I)
) XNDEV(I)=XDEV(I)
  DD=5000.
CULATE HASOFER-LIND SAFETY INDEX - FIRST ESTIMATION.
  CALL OPTM(ZX,AL,BE,N,BETA,IOPT,EE,DD,XR,1.0E0)
LL=0 -- RACKWITZ-FIESSLER METHOD.; IF LL=1 -- LEAST-SQUARES METHOD.
  KK=0
  KK=KK+1
  DO 10 J=1,N
  R-F TRANSFORMATION (LL.EQ.0)
  IF(LL.NE.0) GO TO 50
  IF(DIST(J).NE.2.) GO TO 99
  XNMEAN(J)=XMEAN(J)
  XNDEV(J)=XDEV(J)
  GO TO 10
  CALL CDFPDF(ZX,AL(J),BE(J),DIST(J),DX(J),XMEAN(J),XDEV(J),
 +  0,CDF,PDF)
  U=XINV(CDF)
    STARW=(-(U**2)*.5)
    IF(STARW.LE.-200.) STARW=-200.
  XNDEV(J)=SPI2*EXP(STARW)/PDF
  XNMEAN(J)=DX(J)-U*XNDEV(J)
  GO TO 10
  LEAST SQUARES METHOD (LL.NE.0)
    CONTINUE
  CONTINUE
  IF(LL.NE.0) GO TO 111
  ZBETA=BETA
  CALL OPTM(ZX,AL,BE,N,BETA,IOPT,EE,DD,XR,1.0E0)
 T MAX. NO. OF ITERATIONS FOR DESIGN POINT SEARCH**
  IF(KK.EQ.100) RETURN
  T STOP CRITERIAS FOR THE CALCULATION OF BETA**
  IF(BETA.LT.4.0) GO TO 20
  ERRPER=100.*ABS(BETA-ZBETA)/ZBETA
  IF(ERRPER.LE.0.1) RETURN
  GO TO 77
  IF(ABS(BETA-ZBETA).LE.EPS) RETURN
  GO TO 77
  RETURN
  END


  SUBROUTINE MINBT(ZX,AL,BE,N,BETA)
   SUBROUTINE COMPUTES THE MINIMUM DISTANCE.
    IMPLICIT REAL (A-H,O-Z)
  DIMENSION DIST(20),X(20),XR(20),XMEAN(20),XDEV(20),ZX(20)
  DIMENSION CI(20),AI(20),SF(20),C(20,2),FORM(20),AL(20),BE(20)
  COMMON /OP1/ DIST,X,XR,XMEAN,XDEV,CI,AI,SF,C,FORM,II,ZO,XR1
```

```
      SUM=0.0
C FOR II=0 (R-F) LOOP ONLY.
C COMPUTE BASIC VALUES FROM THE REDUCED VALUES.
      DO 1 I=2,N
      X(I)=XDEV(I)*XR(I-1)+XMEAN(I)
    1 IF(X(I).LE.(0.001*XDEV(I)).AND.DIST(I).NE.2.) X(I)=0.001*XDEV(I)
C COMPUTE X(1) VALUE.
      CALL LSFFPI(N,X,ZX,ZO)
C SET LOWER LIMIT OF THE DESIGN POINT VALUE OF X(M)
      XR1=(X(1)-XMEAN(1))/XDEV(1)
        DO 33 I=1,N
        IF(XR(I).GT.27.) XR(I)=27.
        IF(XR(I).LT.-27.) XR(I)=-27.
33      CONTINUE
        M=N-1
      DO 3 I=1,M
    3 SUM=SUM+XR(I)**2
        SUM=SUM+XR1**2
      BETA=SQRT(SUM)
      RETURN
      END



      SUBROUTINE CDFPDF(ZX,ALPHA,BETA,DIST,X,XMEAN,XDEV,ICDF,CDF,PDF)
C THIS SUBROUTINE CALCULATES THE CDF AND PDF OF THE FOLLOWING
C DISTRIBUTIONS: 1.=WEIBULL,2.=NORMAL,3.=EVD,4.=LOGNORMAL.,5.=FRECHET
C FOR ADDITIONAL DISTRIBUTIONS, THE CDF AND THE PDF MUST BE EXPRESSED
C IN TERMS OF THE MEAN(XMEAN) AND THE STANDARD DEVIATION(XDEV).
        IMPLICIT REAL (A-H,O-Z)
        DIMENSION ZX(20)
        COMMON /TWO/ PI,PI2,SPI2
C SET LOWER LIMIT FOR NON-NORMAL VARIABLES.
      XL=0.00001*XDEV
      IF(DIST.NE.2. .AND. X.LE.XL) X=XL
      GO TO (1,2,3,4,5),INT(DIST+.1E0)
    1 IF(ABS(X).LE.1.E-10) THEN
      CDF=0.E0
      PDF=0.E0
      GO TO 10
      END IF
      RB=X/BETA
      EW=RB**ALPHA
      IF(EW.GT.200.) EW=200.
      EXPWEI=EXP(-EW)
      CDF=1.0E0-EXPWEI
      IF(ICDF.EQ.1) GO TO 10
      PDF=(ALPHA/BETA)*(EW/RB)*EXPWEI
      GO TO 10
    2 Z=(X-XMEAN)/XDEV
      CDF=CDFNOR(Z)
      IF(ICDF.EQ.1) GO TO 10
      PDF=SPI2*EXP(-Z**2*.5)/XDEV
      GO TO 10
    3 E1=ALPHA*(X-BETA)
      IF(E1.GT.200.) E1=200.
      YY=EXP(-E1)
      IF(YY.GT.200.) YY=200.
```

```
CDF=EXP(-YY)
IF(ICDF.EQ.1) GO TO 10
EY=E1+YY
IF(EY.GT.200.) EY=200.
PDF=ALPHA*EXP(-EY)
GO TO 10
CX21=(XDEV/XMEAN)**2+1.
YMEAN=LOG(XMEAN)-LOG(SQRT(CX21))
YDEV=SQRT(LOG(CX21))
Z=(LOG(X)-YMEAN)/YDEV
CDF=CDFNOR(Z)
IF(ICDF.EQ.1) GO TO 10
EZ=-(Z**2)*.5
IF(EZ.LE.-200.) EZ=-200.
PDF=SPI2*EXP(EZ)/(YDEV*X)
GO TO 10
  IF(ABS(X).LE.1.E-10) THEN
  CDF=0.E0
  PDF=0.E0
  GO TO 10
  END IF
  TEMP=(BETA/X)**ALPHA
  CDF=EXP(-TEMP)
  IF(ICDF.EQ.1) GO TO 10
  PDF=CDF*TEMP*ALPHA/X
RETURN
END


SUBROUTINE OPTM(ZX,AL,BE,NP,EF,NFCC,E,ESCALE,X,OPTMIZ)

; IS THE OPTIMIZATION ROUTINE FOR FINDING THE R-F SAFETY INDEX,
THE THREE PARAMETERS OF THE  EQUIVALENT NORMAL CDF.
DIMENSION OF W = NO. OF VARIABLES*(NO. OF VARIABLES + 3)
  IMPLICIT REAL (A-H,O-Z)
DIMENSION X(20),W(460),E(20),ZX(20),AL(20),BE(20)
DIMENSION DIST(20),DX(20),XR(20),XMEAN(20),XDEV(20),CI(20),AI(20),
SF(20),BB(81),CDFNON(81),WEIGT(81),SQRWGT(81),C(20,2),FORM(20)
COMMON /OP1/ DIST,DX,XR,XMEAN,XDEV,CI,AI,SF,C,FORM,II,ZO,XR1
COMMON /OP2/ BB,CDFNON,WEIGT,SQRWGT,DP,NA,NB,PT,SCALE
R INTERMEDIATE RESULTS SET IPRINT TO A LOWER INTEGER.**
IPRINT=4
N=NP-1
ODMAG=0.1*ESCALE
SCER=0.05/ESCALE
JJ=N*(N+1)
JJJ=JJ+N
(=N+1
NFCC=1
END=1
NN=1
)0 4 I=1,N
(I)=ESCALE
)0 4 J=1,N
'(K)=0.
F(I-J)4,3,4
(K)=ABS(E(I))
=K+1
TERC=1
```

```
      ISGRAD=2
      CALL MINBT(ZX,AL,BE,NP,F)
      FKEEP=2.*ABS(F)
    5 ITONE=1
      FP=F
      SUM=0.
      IXP=JJ
      DO 6 I=1,N
      IXP=IXP+1
    6 W(IXP)=X(I)
      IDIRN=N+1
      ILINE=1
    7 DMAX=W(ILINE)
      DACC=DMAX*SCER
      DMAG=AMIN1(DDMAG,0.1*DMAX)
      DMAG=AMAX1(DMAG,20.*DACC)
      DDMAX=10.*DMAG
      IF(ITONE-2)70,70,71
   70 DL=0.
      D=DMAG
      FPREV=F
      IS=5
      FA=FPREV
      DA=DL
    8 DD=D-DL
      DL=D
   58 K=IDIRN
      DO 9 I=1,N
      X(I)=X(I)+DD*W(K)
    9 K=K+1
      CALL MINBT(ZX,AL,BE,NP,F)
      NFCC=NFCC+1
      GO TO (10,11,12,13,14,96),IS
   14 IF(F-FA)15,16,24
   16 IF(ABS(D)-DMAX) 17,17,18
   17 D=D+D
      GO TO 8
   18 CONTINUE
        WRITE(6,19)
   19 FORMAT(5X,44HMAXIMUM CHANGE DOES NOT ALTER FUNCTION(OPTM))
      GO TO 20
   15 FB=F
      DB=D
      GO TO 21
   24 FB=FA
      DB=DA
      FA=F
      DA=D
   21 IF(ISGRAD-1)83,83,23
   23 D=DB+DB-DA
      IS=1
      GO TO 8
   83 D=0.5*(DA+DB-(FA-FB)/(DA-DB))
      IS=4
      IF((DA-D)*(D-DB))25,8,8
   25 IS=1
      IF(ABS(D-DB)-DDMAX)8,8,26
   26 D=DB+SIGN(DDMAX,DB-DA)
      IS=1
```

```
      DDMAX=DDMAX+DDMAX
      DDMAG=DDMAG+DDMAG
      IF(DDMAG.GE.1.0E60) DDMAG=1.0E60
      IF(DDMAX-DMAX)8,8,27
    7 DDMAX=DMAX
      GO TO 8
    6 IF(F-FA)28,23,23
    5 FC=FB
      DC=DB
    4 FB=F
      DB=D
      GO TO 30
      IF(F-FB)28,28,31
      FA=F
      DA=D
      GO TO 30
      IF(F-FB)32,10,10
      FA=FB
      DA=DB
      GO TO 29
      DL=1
      DDMAX=5.
      FA=FP
      DA=-1.
      FB=FHOLD
      DB=0.
      D=1.
      FC=F
      DC=D
      A=(DB-DC)*(FA-FC)
      B=(DC-DA)*(FB-FC)
      IF((A+B)*(DA-DC))33,33,34
      FA=FB
      DA=DB
      FB=FC
      DB=DC
      GO TO 26
      D=0.5*(A*(DB+DC)+B*(DA+DC))/(A+B)
      DI=DB
      FI=FB
      IF(FB-FC)44,44,43
      DI=DC
      FI=FC
      IF(ITONE-2)86,86,85
      ITONE=2
      GO TO 45
      IF(ABS(D-DI)-DACC)41,41,93
      IF(ABS(D-DI)-0.03*ABS(D))41,41,45
      IF((DA-DC)*(DC-D))47,46,46
      FA=FB
      DA=DB
      FB=FC
      DB=DC
      GO TO 25
      IS=2
      IF((DB-D)*(D-DC))48,8,8
      IS=3
      GO TO 8
      F=FI
```

```
      D=DI-DL
      DD=SQRT((DC-DB)*(DC-DA)*(DA-DB)/(A+B))
      DO 49 I=1,N
      X(I)=X(I)+D*W(IDIRN)
      W(IDIRN)=DD*W(IDIRN)
   49 IDIRN=IDIRN+1
      W(ILINE)=W(ILINE)/DD
      ILINE=ILINE+1
      IF(IPRINT-1)51,50,51
   50 IF(IPRINT.GE.4) GO TO 53
      WRITE(6,52) ITERC,NFCC,F,(X(I),I=1,N)
   52 FORMAT(11H  ITERATION,I3,I8,16H FUNCTION VALUES,5X,2HF=,E13.6,
     +5(E13.5,2X))
      IF(IPRINT-1)51,51,53
   51 IF(ITONE-1)55,55,38
   55 IF(FPREV-F-SUM)94,95,95
   95 SUM=FPREV-F
      JIL=ILINE
   94 IF(IDIRN-JJ)7,7,84
   84 IF(IND-1)92,92,72
   92 FHOLD=F
      IS=6
      IXP=JJ
      DO 59 I=1,N
      IXP=IXP+1
   59 W(IXP)=X(I)-W(IXP)
      DD=1.
      GO TO 58
   96 IF(IND-1)112,112,87
  112 IF(FP-F)37,37,91
   91 D=2.*(FP+F-2.*FHOLD)/(FP-F)**2
      IF(D*(FP-FHOLD-SUM)**2-SUM) 87,37,37
   87 J=JIL*N+1
      IF(J-JJ)60,60,61
   60 DO 62 I=J,JJ
      K=I-N
   62 W(K)=W(I)
      DO 97 I=JIL,N
   97 W(I-1)=W(I)
   61 IDIRN=IDIRN-N
      ITONE=3
      K=IDIRN
      IXP=JJ
      AAA=0.
      DO 67 I=1,N
      IXP=IXP+1
      W(K)=W(IXP)
      IF(AAA-ABS(W(K)/E(I)))66,67,67
   66 AAA=ABS(W(K)/E(I))
   67 K=K+1
      DDMAG=1.
      W(N)=ESCALE/AAA
      ILINE=N
      GO TO 7
   37 IXP=JJ
      AAA=0.
      F=FHOLD
      DO 99 I=1,N
      IXP=IXP+1
```

```
 X(I)=X(I)-W(IXP)
 IF(AAA*ABS(E(I))-ABS(W(IXP)))98,99,99
 AAA=ABS(W(IXP)/E(I))
 CONTINUE
 GO TO 72
 AAA=AAA*(1.+DI)
 IF(IND-1)72,72,106
 IF(IPRINT-2)53,50,50
 IF(IND-1)109,109,88
 IF(AAA-0.1)20,20,76
 IF(F-FP)35,78,78
 CONTINUE
   WRITE(6,80)
 FORMAT(5X,37HACCURACY LIMITED BY ERRORS IN F(OPTM))
 GO TO 20
 IND=1
 DDMAG=0.4*SQRT(ABS(FP-F))
 IF(DDMAG.GE.1.E+30) DDMAG=1.0E+30
 ISGRAD=1
 ITERC=ITERC+1
ET MAX. NO. OF ITERATIONS.**
 MAXIT=100
 IF(ITERC-MAXIT)5,5,81
 CONTINUE
   WRITE(6,82) MAXIT
 FORMAT(I5,29H ITERATIONS COMPLETED BY OPTM)
 IF(F-FKEEP)20,20,110
 F=FKEEP
 DO 111 I=1,N
 JJJ=JJJ+1
 X(I)=W(JJJ)
 GO TO 20
 IF(AAA-0.1)20,20,107
 EF=F
 GO TO 666
 INN=1
 GO TO 35
 RETURN
 END
   SUBROUTINE HARBIFN(X,K,ALPHA,BETA,G)
   IMPLICIT REAL (A-H,O-Z)
   DIMENSION X(K)

   G IS PERFORMANCE FUNCTION
   G MUST BE EQUAL TO OR L.T. ZERO TYPE

   G=X(1)-SQRT(3.E2*X(2)**2+1.92EO*X(3)**2)
   RETURN
   END
   SUBROUTINE LSFFPI(N,X,ZX,ZO)
   IMPLICIT REAL (A-H,O-Z)
   DIMENSION X(20),ZX(20)

   PERFORMANCE FUNCTION (X(1)=f(X(2),...,X(N)))

   X(1)=SQRT(3.E2*X(2)**2+1.92EO*X(3)**2)
   RETURN
   END
```

The limit state is

$$g(X) = R - \sqrt{300P^2 + 1.92\,T^2}$$

Note how it is entered into the two subroutines.

Example Problem

$$g(\underset{\sim}{X}) = R - \sqrt{300P^2 + 1.92T^2}$$

$\mu_R = 48$ $\mu_P = 1.0$ $\mu_T = 20$

$\sigma_R = 3$ $\sigma_P = 0.16$ $\sigma_T = 2$

R ~ NEI P ~ LN T ~ EVD

This is an example of the input file

```
THIS IS EXAMPLE 7
1.D-4,3,10000,0.
R
1.,48.,3.
P
4.,.987440632,.16 *
T
3.,20.,2.
```

*Note that P is lognormal; thus the median

$$\tilde{P} = \mu_P / \sqrt{1 + C_P^2}$$

is entered.

THIS IS EXAMPLE 7

DESIGN VARIABLES

VARIABLE	DISTRIBUTION	MEAN/MEDIAN	STD/COV
R	WEIBULL	4.8000E+01	3.0000E+00
P	LOG	9.8744E-01	1.6000E-01
T	EVD	2.0000E+01	2.0000E+00

(NOTE: THE MEDIAN AND COV USED FOR LN)

BETA (SPHERE) = 3.085

NUMBER OF VARIABLES = 3

AREA RATIO, AR = .9934

ALPHA = 2.1880

NUMBER OF SAMPLES = 10000

TOTAL NUMBER OF G < 0 = 798

TOTAL NUMER OF POINTS SAMPLED = 10060

PROBABILITY OUTSIDE BETA SPHERE = 2.31808E-02

PROBABILITY OF FAILURE = 1.84983E-03

BETA = 2.90271

95 CONFIDENCE INTERVAL ON PF

LOWER = 1.72671E-03

UPPER = 1.97294E-03

CPU EXECUTION TIME (SEC.) = 4.28

```
/87  UNIV OF ARIZONA     NOS/BE 1.5 650 87149
.35.SJ  TORNG7U  FROM **    07/06/87
.35.CD  00001015 CARDS,  COST =       $.00
.35.JOBCARD-TORNG,BN4053342 ,T1500.
.35.PW,
.35.FTN5(L=0).
```

APPENDIX D

Particular Solutions for BEM Body Force

Dr. S.T. Raveendra

Southwest Research Institute

For centrifugal loading,

$$P_i = A_1\left[\left(R_{ij}x_j x_k n_k + A_2 R_{jk}x_j x_k n_i\right) + A_3 R_{ij}x_j x_k n_k\right] \qquad \text{(D-1)}$$

where,

$$A_1 = \frac{1}{2\pi\mu}$$

$$A_2 = -\frac{1}{2(1-\nu)}$$

and

$$A_3 = -\frac{(1-2\nu)}{(1-2\nu)}$$

A generalized function for K is selected such that it is global in nature and also the evaluation of the function is computationally efficient. The function selected in this analysis is

$$K(P,Q_m) = R_o\left(1 - \frac{r(P,Q_m)}{R_o}\right) \qquad \text{(D-2)}$$

where $r(P,Q_m)$ is the distance between P and Q_m, and R_o is a characteristic length based on the problem dimensions. The particular solutions presented are based on this function. For thermal analysis

$$G_i^{\theta} = \frac{R_o}{12}\left(4 - \frac{3r}{R_o}\right)y_i \qquad \text{(D-3)}$$

The one-dimensional temperature field solution is given by

$$\psi = \frac{\beta\theta_o(L-\xi)^4}{\left(2L\xi - L^2\right)}e^{-\frac{t}{(L-t)}} \qquad \text{(D-4)}$$

where θ_o is the surface temperature, and

$$\beta = \left(\frac{1+\nu}{1-\nu}\right)\alpha$$

The temperature dependent material properties solution is given by

$$G_{ijk}^{i} = \left(C_1 + rC_2\right)\left(\delta_{jk}y_i + \delta_{ik}y_j + \delta_{ij}y_k\right) + C_2\frac{y_i y_j y_k}{r} \qquad \text{(D-5)}$$

where

$$C_1 = \frac{R_o}{15(\lambda+2\mu)}$$

and

$$C_2 = -\frac{1}{24(\lambda+2\mu)}$$

The vibration analysis particular solutions are given by

$$G^d_{ij} = C_3(\delta_{ij}r^2 + y_i y_j) + C_4\delta_{ij}r^3 + C_5 r y_i y_j \qquad (I$$

$$T^d_{ij} = \mu\left[(3C_3 + rC_6)y_j n_i + (C_7 + rC_8)y_i n_j + \left\{(3C_3 + rC_6)\delta_{ij} + 2C_5\frac{y_i y_j}{r}\right\}y_k n_k\right] \qquad (I$$

where

$$C_3 = -\frac{(1-2\nu)R_o}{(14-16\nu)\mu}$$

$$C_4 = \frac{11-12\nu}{144(1-\nu)\mu}$$

$$C_5 = -\frac{1}{48(1-\nu)\mu}$$

$$C_6 = -\frac{5-6\nu}{24(1-\nu)\mu}$$

$$C_7 = -\frac{(1+4\nu)}{(7-8\nu)\mu}R_o$$

and

$$C_8 = \frac{1-6\nu}{24(1-\nu)\mu}$$

The unknown coefficients are obtained from

$$\underline{F}^\theta = \underline{K}^{-1} \qquad (D$$

$$\underline{F}^i = \underline{K}^{-1} \qquad (D$$

$$\underline{F}^d = \underline{K}^{-1} \qquad (D$$

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

1. AGENCY USE ONLY (*Leave blank*)	2. REPORT DATE	3. REPORT TYPE AND DATES COVERED
	October 1991	3rd Annual Contractor Report Oct. 1987

4. TITLE AND SUBTITLE
Probabilistic Structural Analysis Methods (PSAM) for Select Space Propulsion System Components
(3rd Annual Report)

5. FUNDING NUMBERS
WU-553-13-00

6. AUTHOR(S)

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
Southwest Research Institute
6220 Culebra Road
San Antonio, Texas 78284

8. PERFORMING ORGANIZATION REPORT NUMBER
None

9. SPONSORING/MONITORING AGENCY NAMES(S) AND ADDRESS(ES)
National Aeronautics and Space Administration
Lewis Research Center
Cleveland, Ohio 44135-3191

10. SPONSORING/MONITORING AGENCY REPORT NUMBER
NASA CR-187197

11. SUPPLEMENTARY NOTES
Project Manager, C.C. Chamis, Structures Division, NASA Lewis Research Center, (216) 433-3252.

12a. DISTRIBUTION/AVAILABILITY STATEMENT
Unclassified - Unlimited
Subject Category 39

12b. DISTRIBUTION CODE

13. ABSTRACT *(Maximum 200 words)*

This Annual Report summarizes the work completed during the third year of technical effort on the referenced contract. Principal developments continue to focus on the Probabilistic Finite Element Method (PFEM) which has been under development for three years. Essentially all of the linear capabilities within the PFEM code are in place. Major progress in the application or verifications phase has been achieved. An EXPERT module architecture has been designed and partially implemented. EXPERT is a user interface module which incorporates an expert system shell for the implementation of a rule-based interface utilizing the experience and expertise of the user community. The Fast Probability Integration (FPI) algorithm continues to demonstrate outstanding performance characteristics for the integration of probability density functions for multiple variables. Additionally, an enhanced Monte Carlo simulation algorithm has been developed and demonstrated for a variety of numerical strategies.

14. SUBJECT TERMS
Structure; Materials; Finite elements; Displacements; Eigenvalues; Transient; Harmonic; Vibration; Beams; Plates; Shells; Expert systems; Correlated fields

15. NUMBER OF PAGES
418

16. PRICE CODE
A18

17. SECURITY CLASSIFICATION OF REPORT	18. SECURITY CLASSIFICATION OF THIS PAGE	19. SECURITY CLASSIFICATION OF ABSTRACT	20. LIMITATION OF ABSTRACT
Unclassified	Unclassified	Unclassified	